AFIT/GA/AA/78D-8

ERROR PROPAGATION FOR GROUND-TO-
SATELLITE ELLIPTICAL INTERCEPT ORBITS
TO EXAMINE THE ERROR ELLIPSOID GROWTH.

THESIS

AFIT/GA/AA/78-D-8 Michael J. VonPlinsky
2nd. Lt.   USAF

A

AFIT/GA/AA/78D-8

ERROR PROPAGATION FOR GROUND-TO-SATELLITE ELLIPTICAL

INTERCEPT ORBITS TO EXAMINE THE ERROR ELLIPSOID GROWTH

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air Training Command

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science.

By:

Michael J. Von Plinsky, B.S.

2nd Lt.   USAF

Graduate Astronautical Engineering

December 1978.

Preface

In selecting the topic for this study, I was moti-
vated by a want to apply the span of knowledge I have
acquired at AFIT and by the fact that I wanted to re-
search a current problem which would lend experience
and insight toward future possible assignments.

A number of people were instramental in the comp-
letion of this study. First and foremost, I would like
to express my sincere appreciation to my thesis asvis-
or, Captain William E. Wiesel. Without his expert ad-
vice and timely aid this study would have not been pos-
sible. I would also like to thank the other two members
of my thesis committee, Dr. Calico and Captain Rader,
for their concise and helpful comments on the final
preparation of this report. Finally, I would like to
thank my typist, Denise, whose perseverence throughout
was greatly appreciaed.

Michael J. Von Plinsky

Table of Contents

Table of Contents

List of Figures

List of Tables.

AFIT/GA/AA/78D-8

Abstract

Errors from the intended performance of a system,
composed of specialized co. ponents, could be due to any
number of sub-system fluctuations. This thesis will
examine the effect of these resultant errors on the
mission of the system involved. In particular, errors
in an interceptor's booster performance (seen as position
and velocity errors at burnout) will be propagated along
the interceptor's orbit to a given target point and the
resultant error at the intercept point will be found. A
range of intercept orbits will be examined to determine if
there is a family of particular intercept orbits that give
the minimum position errors at the intercept point. Con-
clusions will be made as to what typical intercept trajec-
tory should be flown against a certain type of target so
as to minimize the resultant free flight error propagation
to the intercept point.

I Introduction

Background

In the problem of intercepting any orbital target, there are always inherent abberations in the interceptor system components which cause deviations from the desired trajectory during the course of the intercept. The unplanned errors in the flight path could be due to any number of factors which could effect the targeted vehicle in reaching it's programmed orbit. These deviations are caused by unavoidable design and manufacturing flaws in the components that compose the missle booster system, such as fluctuations in the burn of the rocket motor during orbital insertion or small errors caused by a slightly inaccurate guidance package. All of these factors contribute to position and velocity errors which occur during the powered phase of the mission, $i\ell$ - up to booster burnout. Hence, the errors generated during the powered portion of the flight will be the ultimate causes of errors in position and velocity of the interceptor at it's planned target point termination.

Objectives

This study will be concerned with investigating how
the inherent errors of a missle intercept system effect
the final targeting of the interceptor involved. Depending
on the capability of the booster svstem, each interceptor
has the versatility to reach an orbital target using a
variety of trajectories, with each programmed to intercept
the body at a different point in it's orbit. Knowing the
accuracy characteristics of the booster system being used
allows the propagation of each possible intercept trajectory,
over a variety of times-of-flight, which give an elliptical
intercept path. Once the flight path is traced to it's
termination at the target's orbit, the deviation from the
proposed intercept point can be found by comparing the ideal
(no error) orbit and corresponding intercept point to the
actual orbit found through propagation of the error
ellipsoid from booster burnout to the target's orbital
path. Thus, given an error covariance matrix modeling the
position and velocity errors at burnout, this covariance
matrix can be propagated by means of a state transition
matrix to the predicted intercept point to give a covar-
iance at intercept. Then, examination of the eigenvalues
and eigenvectors of this final covariance matrix results
in the generation of an error ellipsoid for the interceptor

at the desired target point. This error ellipsoid can
then be sliced appropriatly to indicate deviations from
the intended intercept point in the cross track directions
of the target's orbit. This final process gives actual
miss parameters at the intercept point from the target body
in question. Repeating this process over the range of
possible intercept trajectories will give data which can
be examined to find the family of trajectories for which the
minimum cross track intercept errors are achieved.

Research

In my research for previous work, or corresponding
effort on my topic, I found no data or documentation on the
subject. This was to be expected due to the uniqueness of
the problem. I did, however, uncover many items that proved
invaluable in the development of this project. Several
supportive references were found which aided in the extensive
orbital dynamics modeling which was done in the program.
The typical off-the-shelf general references proved invaluable
in helping me answer the "obvious" questions throughout the
project.

II Computer Simulation

Assumptions

A computer program has been developed to simulate the aspects of this problem, as previously outlined. For the purposes of this study, it will be assumed that both the orbital target and the interceptor are under the influence of two-body effects only. Since the typical intercept orbit would probably be an elliptical one, due to launch and booster constraints, only elliptical intercept orbits will be examined to further simplify the process. It is also assumed that the satellite's orbital position at launch is perfectly known; thus, the only errors present at the intercept point will be due to position and velocity deviations of the interceptor. The time between receiving the initial satellite position data and launch is assumed to be zero, and an instantaneous burn time of the interceptor's booster is also assumed. In order to simplify the implementation of the many orbital dynamics formulas needed, a universal variable approach will be used throughout the project.

Programming Considerations

Initially, launch site data will be inputed to find the

position of the launch site in inertial space. Satellite data, in the form of a classical element set for the target satellite in question, will also be inputed to find the initial position and velocity of the satellite, also in inertial space. An initial time-of-flight for the interceptor will be chosen and it's corresponding orbit to the intercept point (the point where the target satellite would be after that time-of-flight had elapsed) will be computed. A test will be made to see if this computed orbit is elliptical. If this orbit is not elliptical, then the time-of-flight will be iterated, and the initial proceedure repeated until an elliptical intercept orbit is found. Once this is accomplished, the state transition matrix for that particular elliptical orbit will be computed. This state transition matrix will be used to propagate a covariance matrix, representing the position and velocity errors of the interceptor at the booster's burnout point, from the burnout point to the given intercept point. Thus, a covariance matrix will be found at the intercept point, representing the propagated position and velocity errors of the interceptor. By examining this final covariance matrix, conclusions can be made as to the effect initial booster launch errors have on the final miss parameters, associated with the interception of the target satellite.

Computing The Launch Site Coordinates In Inertial Space

This initial phase of the program was designed to allow the earth location of any possible site to be entered, as well as the date and time of launch, to allow flexibility in the final simulation. Inputed data consisted of the latitude and longitude of the site, it's elevation above sea level, and the year, day, and time of launch. A subroutine was developed to determine the local sideral time of the launch site given the site's longitude and the year, day, and time of launch (Ref 1:103-104). Then, knowing the latitude, altitude, and local siderial time of the site, a subroutine was developed to determine the position of the site in the topocentric reference frame. This position vector was then transformed by means of an appropriate rotation matrix to give the position vector of the launch site in the inertial frame (see Fig 1), (Ref 1:98).

Computing The Satellite's Initial Position In Inertial Space

In computing the satellite's position at epoch, or launch, data from a typical element set, which would be received at the launch site from the NORAD space track system, was used. This was composed of the length of the semi-major axis of the satellite's orbit, it's orbital eccentricity, it's inclination, the longitude of the

Figure 1.   Relationship Between S-E-Z and I-J-K

ascending node, the arguement of perapsis, and it's true
anomaly $(A, \ell, i, \Omega, \omega, \nu)$, (see Fig 2.). Knowing these para-
meters, the initial position and velocity of the satellite
can be expressed in vector form in the perifocal coordinate
system by the following equations (ref 1, 72):

$$\bar{r} = |r| \cos \nu \, \bar{P} + |r| \sin \nu \, \bar{Q} \qquad (1)$$

$$\bar{v} = \sqrt{\mu/p} \left[-\sin \nu \, \bar{P} + (e + \cos \nu) \bar{Q} \right] \qquad (2)$$

where

$$P = A(1 - e^2)$$

$$|r| = P/(1 + e \cos \nu)$$

and in universal variables $\mu = 1.$

7
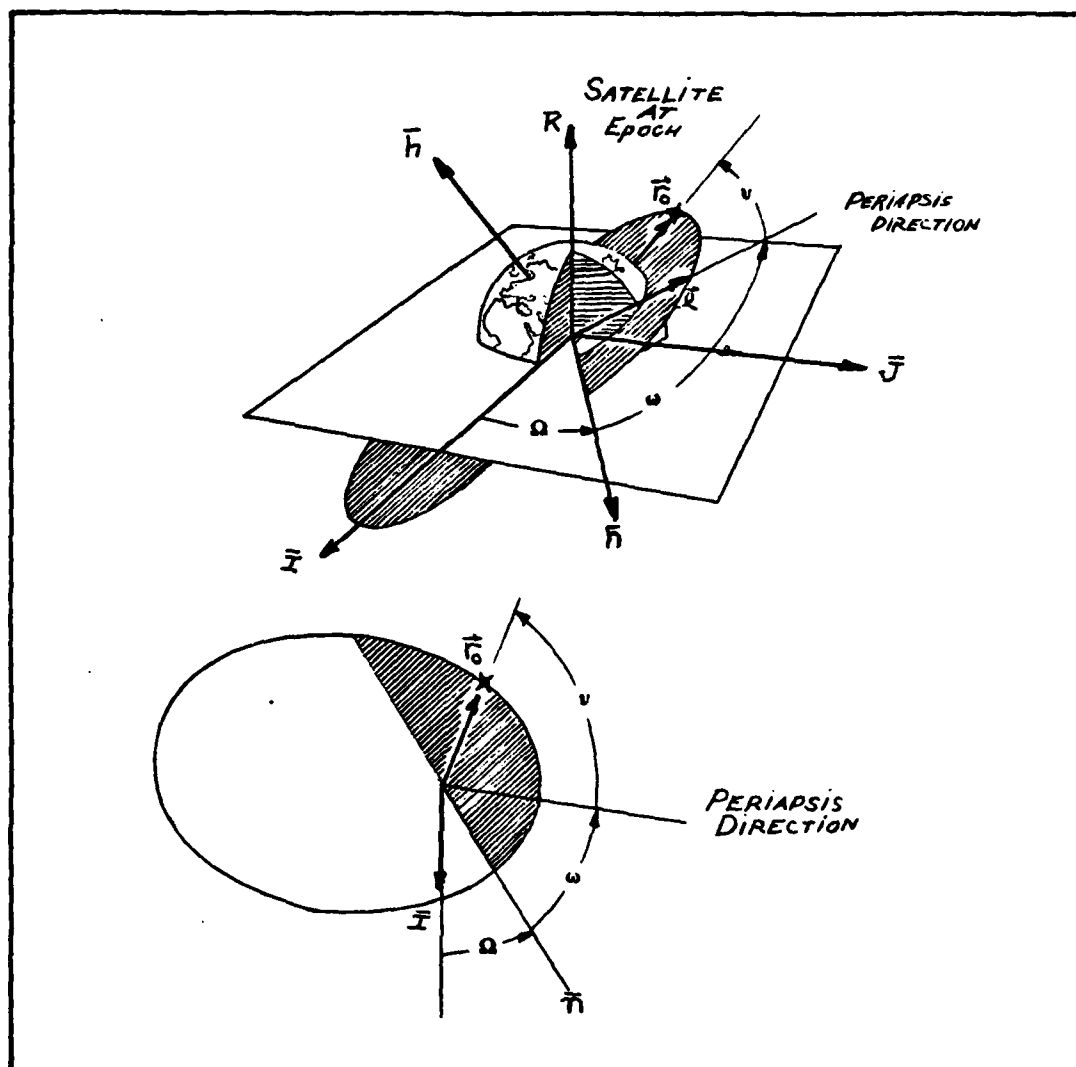
Figure 2.   Orbital Elements

The transformation of coordinates between the perifocal
system (P,Q,W) and the inertial system (I,J,K) can be
accomplished by means of a rotation matix $\hat{R}$ (Ref 1:82).
Thus, if $(\bar{a}_p, \bar{a}_q, \bar{a}_w)$ are the components of a vector $\bar{a}$ in the
perifocal frame, then the coordinates of that vector in the
inertial frame are given by:

8

$$\begin{bmatrix} \bar{a}_I \\ \bar{a}_J \\ \bar{a}_K \end{bmatrix} = \widetilde{R} \begin{bmatrix} \bar{a}_P \\ \bar{a}_Q \\ \bar{a}_W \end{bmatrix} \tag{3}$$

where:

$$\widetilde{R} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{4}$$

$R_{11} = \cos\Omega \cos\omega - \sin\Omega \sin\omega \cos i$

$R_{12} = -\cos\Omega \sin\omega - \sin\Omega \cos\omega \cos i$

$R_{13} = \sin\Omega \sin i$

$R_{21} = \sin\Omega \cos\omega + \cos\Omega \sin\omega \cos i$

$R_{22} = -\sin\Omega \sin\omega + \cos\Omega \cos\omega \cos i$

$R_{23} = -\cos\Omega \sin i$

$R_{31} = \sin\omega \sin i$

$R_{32} = \cos\omega \sin i$

$R_{33} = \cos i$

Having determined the elements of the rotation matrix, it only remains to find $\vec{r}$ and $\vec{v}$ in terms of the inertial frame components (I, J, K). Thus:

$$\begin{bmatrix} \bar{r}_I \\ \bar{r}_J \\ \bar{r}_K \end{bmatrix} = \widetilde{R} \begin{bmatrix} \bar{r}_P \\ \bar{r}_Q \\ \bar{r}_W \end{bmatrix} \quad \text{AND} \quad \begin{bmatrix} \bar{v}_I \\ \bar{v}_J \\ \bar{v}_K \end{bmatrix} = \widetilde{R} \begin{bmatrix} \bar{v}_P \\ \bar{v}_Q \\ \bar{v}_W \end{bmatrix} \tag{5,6}$$

Hence, the satellite's position and velocity vectors in the inertial frame are now known at epoch (i_R;launch time).

Find The Satellite's Position At The Intercept Point

In computing the orbit needed for intercept, the time-

of-flight for the interceptor was chosen as a variable
which would give a resultant elliptical intercept orbit.
Due to the fact that the launch time was chosen to coincide
with the epoch time for which the initial position of the
target satellite was computed, the intercept point will be
the point on the satellite's orbit where it will be located
after traveling the chosen time-of-flight for the interceptor.
Thus, the first step in computing the intercept orbit is
finding the intercept point. This is done by up-dating the
satellite's position from it's initial location at epoch
($t=o$) to it's resultant location   , marked by the satellite's
travel during the time-of-flight of the interceptor. This
prediction problem can be stated as follows:

$$\text{Given:} \qquad \vec{r_0} \;,\; \vec{v_0} \;,\; t_0 = 0$$

$$\text{Find:} \qquad \vec{r} \;,\; \vec{v} \;\; \text{at time } t$$

where t is the time-of-flight of the interceptor. (see
Fig 3).

The solution consists of solving the Kepler problem for
the satellite as it travels from $\vec{r_0}$ to $\vec{r}$ in the given time-
of-flight. The initial step is to compute the universal
variable X, which is defined by the relation $\dot{X} = 1/r$.
Kepler's time-of-flight equation expressed in universal

Figure 3. The Kepler Problem

variables is soley dependent on X and the initial position
and velocity of the satellite, as seen here:

$$TOF = (\vec{r_0} \cdot \vec{v_0})X^2 C + (1 - \frac{r_0}{A})X^3 S + r_0 X \qquad (7)$$

where:

$$S = (\sqrt{z} - \sin\sqrt{z})/\sqrt{z^3}$$
$$C = (1 - \cos\sqrt{z})/z$$
$$z = X^2/A$$

An intermediate step in finding the radius and velocity
vectors at a future time is to find X when time is known.
Since equation (7) is trancendental in X, a trial and
error solution is indicated. Fortunately, the t vs X

curve is well-behaved and a Newton iteration technique can
be used to solve for X when the time-of-flight is given.
If a trial value for X is chosen, call it X_N, then

$$TOF_N = (\vec{r}_o \cdot \vec{V}_o) X_N^2 C + (1 - r_o/A) X_N^3 S + r_o X_N \qquad (8)$$

where: TOF_N is the time-of-flight corresponding to the
given $\vec{r}_o, \vec{V}_o, A$ and trial value X_N.

A better approximation for X_N is then obtained from the
Newton iteration algorithm:

$$X_{N+1} = X_N + \frac{t - t_N}{dt/dx \,|_{x=x_N}} \qquad (9)$$

where: t is the given time-of-flight and $\frac{dt}{dx}$ is the
slope of the t vs X curve at the trial point X_N. Now,
knowing X we wish to calculate the $\vec{r}$ and $\vec{V}$ vectors in
terms of $\vec{r}_o, \vec{V}_o$ & X. These vectors can be expressed in
terms of scalar quantities dependent on X as follows:

$$\vec{r} = f\, \vec{r}_o + g\, \vec{V}_o \qquad (10)$$
$$\vec{V} = \dot{f}\, \vec{r}_o + \dot{g}\, \vec{V}_o$$

where:

$$f = 1 - (x^2/r_o) C \qquad g = t - x^3 S$$
$$\dot{f} = x(zS - 1)/(r_o r) \qquad \dot{g} = 1 - x^2 c/r$$

Thus, the satellite's position and velocity up-dated by the
chosen time-of-flight for the interceptor is now known.
Having the satellite's position gives us the data for the
interceptor's orbit: the initial position at launch and the
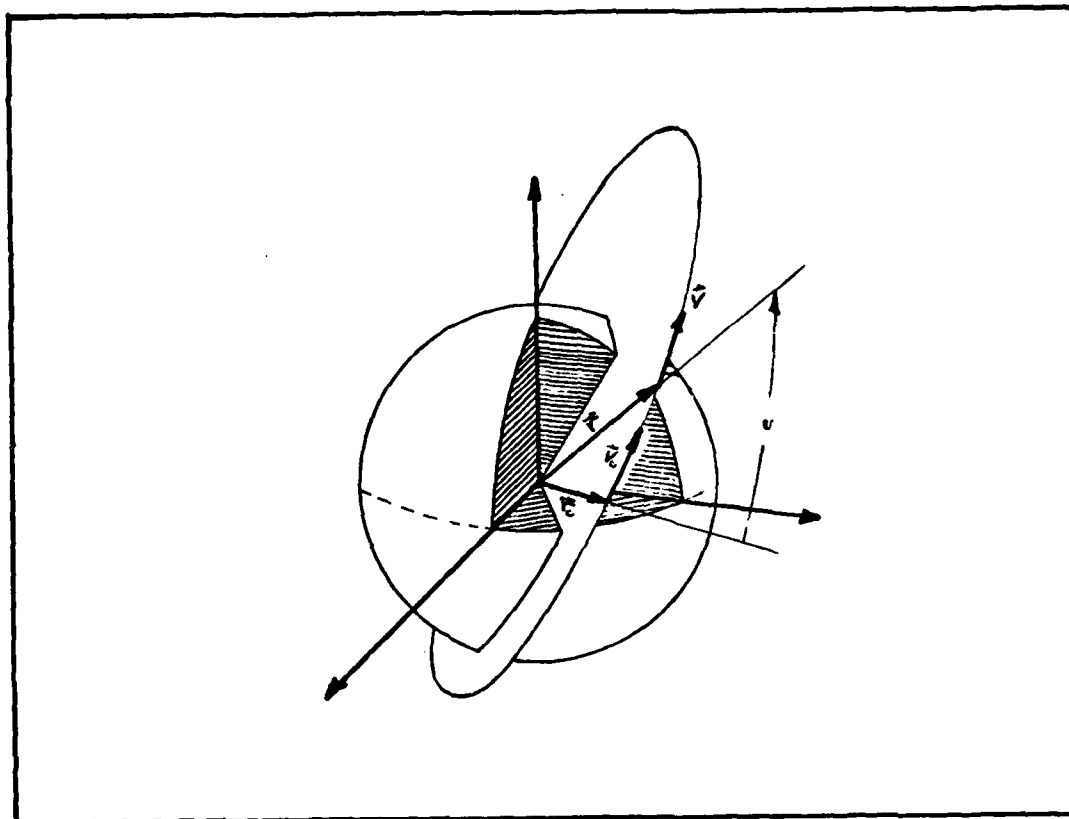final position at intercept.
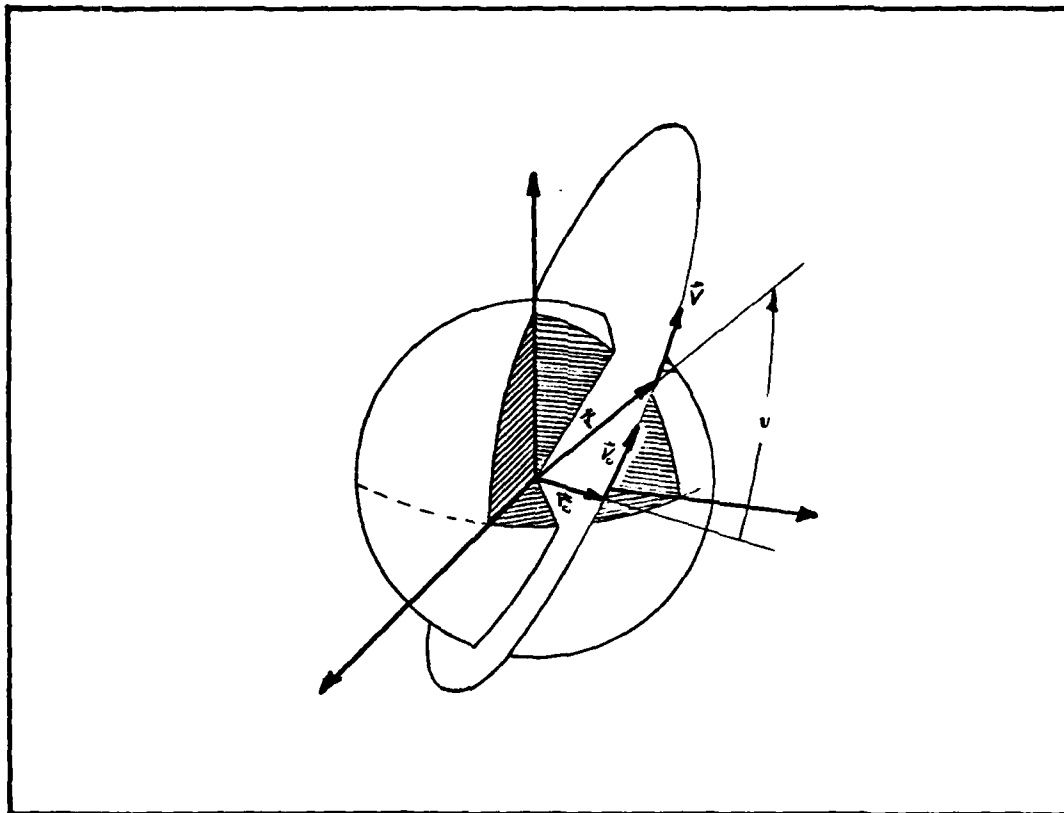
Figure 3. The Kepler Problem

variables is soley dependent on X and the initial position
and velocity of the satellite, as seen here:

$$TOF = (\vec{r_0} \cdot \vec{v_0})X^2 C + (1 - \frac{r_0}{A})X^3 S + r_0 X \tag{7}$$

where:

$$S = (\sqrt{z} - \sin\sqrt{z})/\sqrt{z^3}$$
$$C = (1 - \cos\sqrt{z})/z$$
$$z = X^2/A$$

An intermediate step in finding the radius and velocity
vectors at a future time is to find X when time is known.
Since equation (7) is trancendental in X, a trial and
error solution is indicated. Fortunately, the t vs X

11

Compute The Orbit Between The Launch And Intercept Points

At this point, both the launch point and intercept point of the interceptor are known. It is now necessary to find the orbit between them. In other words, given $\vec{r_1}$, (launch site radius vector), $\vec{r_2}$ (intercept point radius vector) and the time-of-flight from $\vec{r_1}$ to $\vec{r_2}$, find $\vec{V_1}$ and $\vec{V_2}$ where $\vec{V_1}$ is the velocity vector at launch needed to intercept at $\vec{r_2}$ and $\vec{V_2}$ is the interceptor's velocity at $\vec{r_2}$. Here, the time-of-flight is known and is simply the chosen time-of-flight for the interceptor. Thus, the solution of the generalized Gauss problem in universal variables will give the needed orbit. A subprogram was developed to solve the Gauss problem. Initially, from $\vec{r_1}$ and $\vec{r_2}$, the constant A was evaluated, where:

$$A = \frac{\sqrt{r_1 r_2}\ \sin \Delta \nu}{\sqrt{1 - \cos \Delta \nu}} \tag{12}$$

Here, $\Delta \nu$ was evaluated by finding the angle between $\vec{r_1}$ and $\vec{r_2}$. A trial value for z was then chosen and the functions S and C were evaluated for the selected trial value for z where the series representations for S and C were used to eliminate instability problems when z is near zero.

$$S = \frac{1}{2!} - \frac{z}{4!} + \frac{z^2}{6!} - \frac{z^3}{8!} + \ldots \tag{13}$$

$$C = \frac{1}{3!} - \frac{z}{5!} + \frac{z^2}{7!} - \frac{z^3}{9!} + \ldots \tag{14}$$

In the actual program, the first four terms in these series were used.

$$S = \frac{\sqrt{z} - \sin\sqrt{z}}{\sqrt{z^3}} \tag{15}$$

$$C = \frac{1 - \cos\sqrt{z}}{z} \tag{16}$$

Next, the auxiliary variable Y was found from:

$$Y = r_1 + r_2 - A\frac{(1 - zS)}{\sqrt{c}} \tag{17}$$

and X was determined from:

$$X = \sqrt{Y/c} \tag{18}$$

Now, the trial value for z was checked by computing t_N from:

$$t_N = X^3 S + A\sqrt{Y} \tag{19}$$

and it was compared to the desired time-of-flight which is the given time-of-flight chosen for the interceptor. If it is not nearly the same, the trial value of t is obtained, within some convergence criterion. A Newton iteration scheme for the adjustment of z was used, much like that USED IN the PREVIOUS SECTION, and was implemented as follows:

$$Z_{N+1} = Z_N + \frac{t - t_N}{dt/dz\,|_{z=z_N}}$$

where t = desired time-of-flight

t_N is given in equation (19)

$$\frac{dt}{dx} = x^3\left(S' - \frac{3SC'}{2C}\right) + \frac{A}{8}\left(\frac{3S\sqrt{Y}}{C} + \frac{A}{x}\right)$$

$$S' = \frac{1}{2z}(C - 3S)$$

$$C' = \frac{1}{2z}(1 - zS - 2C)$$

Here, again, instability problems could occur for a small value of z. Thus, series expansions for S' and C' were used when z is small and the function representations for S' and C' were used otherwise. When the method has converged to a solution, the functions f, g, and $\dot{g}$ were evaluated from:

$$f = 1 - \frac{Y}{r_1} \tag{23}$$

$$g = A\sqrt{Y} \tag{24}$$

$$\dot{g} = 1 - \frac{Y}{r_2} \tag{25}$$

Now, since $\vec{r_2} = f\vec{r_1} + g\vec{V_1}$, we can compute $\vec{V_1}$ from:

$$\vec{V_1} = \frac{\vec{r_2} - f\,\vec{r_1}}{g} \tag{26}$$

and similarly:

$$\vec{V_2} = \frac{\dot{g}\,\vec{r_2} - \vec{r_1}}{g} \tag{27}$$

Thus now, both the intial position ($\vec{r_1}$) and velocity ($\vec{V_1}$), and the final position ($\vec{r_2}$) and velocity ($\vec{V_2}$) of the interceptor are known, and hence, so is the intercept orbit. (see Fig 4)
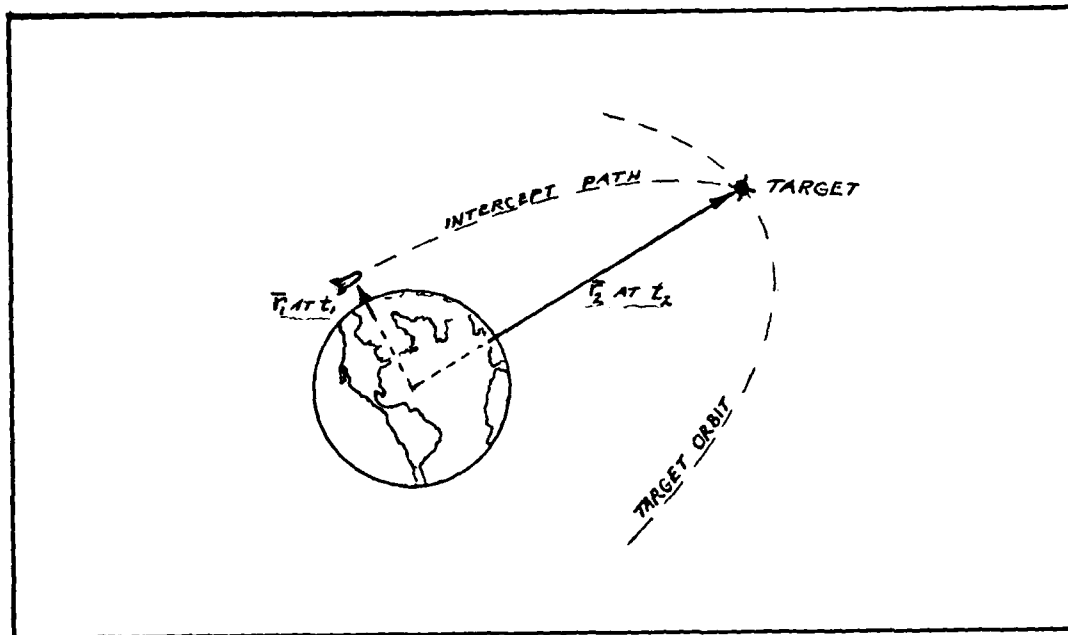
Figure 4.   Intercept Problem

State Transition Matrix

The states at both the launch point and the intercept
point are now completely defined in terms of inertial
position and velocity vectors. The transition matrix
between these two states can now be found by taking the
partials of the state at the intercept time with respect
to the partials of the state at the intercept time with
respect to the partials of the state at the launch time
(Ref 3:129):

$$\Phi\,(t_I, t_0) = \frac{\partial \bar{x}\,(t_{INTERCEPT})}{\partial \bar{x}\,(t_{EPOCH})}\bigg|_{x_0} \tag{28}$$

This was done numerically by perturbing each state element
at launch by a small amount and calculating the resulting

16

new orbit over the same time-of-flight. The difference
between the final perturbed state and the initially known
state at intercept, divided by the amount of the initial
perturbation, gives the elements of the state transition
matrix between the two states, Φ . This transition matrix
will be used to propagate covariances over the known iner-
tial orbit.

Covariance Matrix Propagation

At the burnout point of the booster, a covariance
matrix was developed to reflect the position and velocity
errors due to booster performance. These error parameters
were treated as input to the program. Thus, the position
and velocity sigmas were inputed, transformed from the MKS
to the universal variable system, and squared to give the
diagonal elements of the booster covariance matrix at
burnout, which is itself a diagonal 6 x 6 matrix. The
upper three diagonal elements of the matrix represented the
position sigmas while the lower three diagonal elements
represented the velocity sigmas. Then, using the state
transition matrix for the given intercept orbit, this
covariance matrix was propagated from the burnout point
to the intercept point along the intercept orbit. This
was done via the following equation (Ref 2):

17

$$P_I = \Phi\, P_B\, \Phi^T \tag{28}$$

where:

P_I = COVARIANCE MATRIX AT INTERCEPT

P_B = COVARIANCE MATRIX AT BURNOUT

Φ = STATE TRANSITION MATRIX.

The justification for the use of the above transformation

equation is seen in the following derivation (Ref 3):

$$\delta x(t_I) = \Phi(t_I, t_0)\, \delta x(t_0) \quad \left(\begin{array}{c}\text{DEFINITION of} \\ \text{TRANSITION MATRIX}\end{array}\right) \tag{30}$$

$$P_I = E\left[\delta x(t_I)\, \delta x(t)^T\right] \quad \left(\begin{array}{c}\text{DEFINITION of} \\ \text{COVARIANCE MATRIX}\end{array}\right) \tag{31}$$

$\underline{Thus:}$

$$P_I = E\left[\Phi(t_I, t_0)\, \delta x(t_0)\, \left(\Phi(t_I, t_0)\, \delta x(t_0)\right)^T\right]$$

$$= E\left[\Phi(t_I, t_0)\, \delta x(t_0)\, \delta x(t_0)^T \Phi(t_I, t_0)^T\right]$$

$$= \Phi(t_I, t_0)\, E\left[\delta x(t_0)\, \delta x(t_0)^T\right] \Phi(t_I, t_0)^T$$

$$P_I = \Phi(t_I, t_0)\, P_B\, \Phi(t_I, t_0)^T \tag{32}$$

where: $E[\ ]$ indicates the expected value of the

bracketted quantity. Since this thesis is interested in

only the resultant _position_ _errors_ at intercept, the pro-

pagated covariance matrix was examined and the upper 3 x 3

sub-matrix was extracted. This sub-matrix represents the

propagated position errors at intercept since the initial

covariance matrix was developed by placing the position error elements in the upper diagonal locations of that matrix. This position sub-matrix represents the position errors at intercept with respect to the interceptor's orbit; $\mathcal{L}_e$: the deviations from the intended intercept point referenced from the track of the interceptor's orbital path. However, the items of interest are the deviations from the location of the intended target relative to that target. Thus, the position sub-matrix requires a transformation so as to reflect the actual position errors referenced to the relative velocity vector. This was accomplished by a rotation of the position covariance matrix, rotating about the relative velocity vector between the satellite and the interceptor. This was done by realigning the basis vectors of the error ellipsoid along the relative velocity vector between the interceptor and target satellite (see Fig 5). This rotation matrix was developed as follows (Ref 1:61):

$$R' = \begin{bmatrix} \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ -\sin\phi & \cos\phi & 0 \\ \sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \end{bmatrix} \tag{33}$$
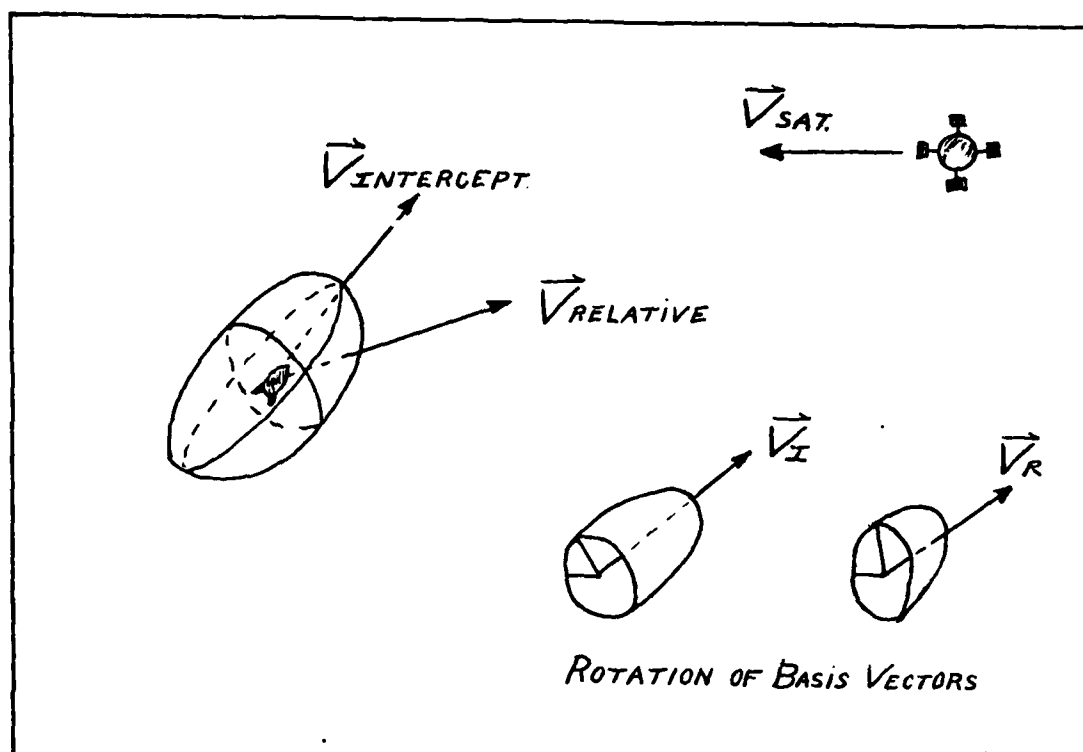
Figure 5. Slice Ellipsoid To Obtain Relative Error Axes

Using this rotation matrix, the position covariance matrix was rotated into the relative velocity vector as follows (Ref 2): Given initial covariance elements $\delta\bar{x}$ and desired rotated covariance elements $\delta\bar{x}'$:

$$\delta\bar{x}' = R' \delta\bar{x} \tag{34}$$

$$P' = E\left[\delta\bar{x}', \delta\bar{x}'^{T}\right] \qquad P = E\left[\delta\bar{x}\,\delta\bar{x}^{T}\right] \tag{35, 36}$$

$$P' = E\left[R'\delta\bar{x}\,(R'\delta x)^{T}\right]$$

$$= E\left[R'\delta x\,\delta x^{T} R'^{T}\right]$$

$$= R'\,E\left[\delta\bar{x}\,\delta\bar{x}^{T}\right]R'^{T}$$

$$P' = R'\,P\,R'^{T} \tag{37}$$

Now the position error ellipsoid at the intercept point
relative to the satellite is known (see Fig 6). It now
remains to find the actual miss distances from this covar-
iance matrix. Due to the intercept problem involved, the
in-track errors, those errors that indicate if the inter-
ceptor is either in front of, or behind the target at
intercept, are of no real interest to this study, for
they simply dictate if the intercept should occur a little
sooner, or later, than expected. Thus, the row and column
of the position covariance matrix that reflect the in-track
errors were eliminated, leaving a 2 x 2 matrix representing
the cross-track errors at intercept. The eigenvalues of
this matrix were then found and their square roots were
taken to determine the axis lengths of the cross-track
error ellipsoid at the intercept point. Once these
principal axis lengths were converted back to the MKS
system, these lengths represented the propagated miss
distances from the target point (in meters). These error
ellipsoid axis lengths, representing the error from the
center of the ellipse to a point on it's edge, are one
sigma errors. They represent a 68% confidence level.
Due to the linearity of the error propagation process,
the three sigma or 99% confidence level can be found by
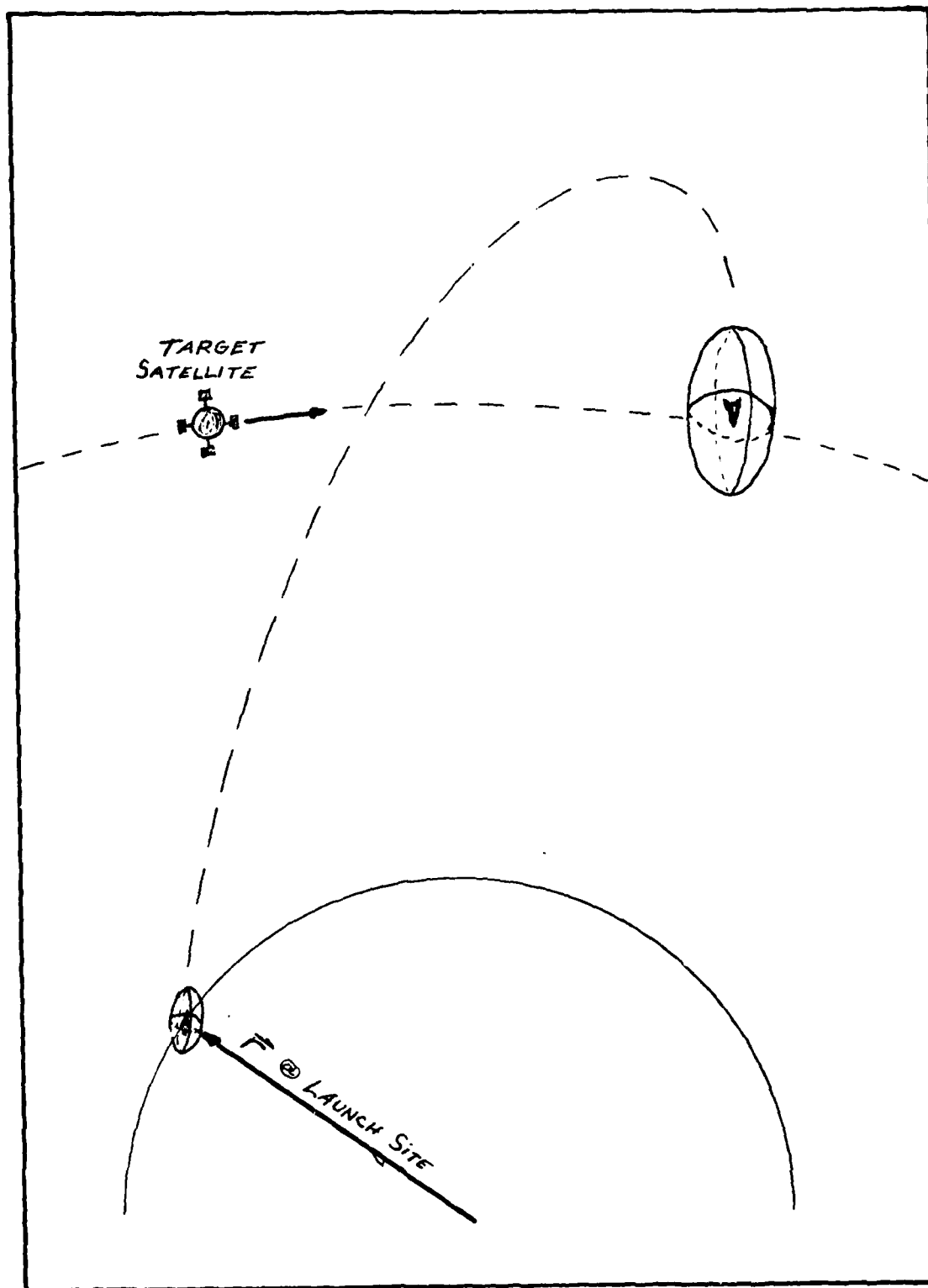simply multiplying the errors by a factor of three.

Figure 6. Error Ellipsoid Growth To Intercept Point

Thus, the error ellipsoid footprint cross-track to the relative velocity vector is now known, along with the magnitudes of each perpendicular arm of the axis which make up that cross-track ellipsoid. Throughout the development of the program, the reliability of each major sub-program or element of the main program was examined in a variety of ways. All of the program elements concerned with orbit determination were verified by using data from examples previously done in the references from which that element was developed. Only high correlation between the output of that sub-program and the "book answer" insured the proper functioning of that program element. The matrix manipulations in the latter part of the main program were verified by examining output matricies during intermediate steps and calculating, by hand, certain elements of the matrix found in the subsequent step and comparing the corresponding numbers. This technique assured the reliability of the intricate matrix manipulations used in the propagation of the initial covariance matrix to the intercept point.

III Program Execution

With the theory now modeled, the program is now ready
to produce the data required from which appropriate con-
clusions can be made. The launch site was held constant,
and a variety of target orbits were examined to make up
one senario. For each orbit examined, varrying times-of-
flight for the interceptor were used until an elliptical
family of intercept trajectories were found. Then the
time-of-flight was slowly iterated over this elliptical
range to produce a quantity of intercept points and, hence,
two-dimensional position error data at each point. This
array of data was examined and the minimum horizontal and
vertical position errors were extracted along with their
corresponding time-of-flight, which indicates on what type
of intercept trajectory they occured. Also, plots of the
cross-track error ellipsoid foot prints were made for the
first and last elliptical orbits in the family and also for
the cases of the minimum horizontal and vertical miss
distances.

Results

In implementing the program to generate data, many
target orbits were examined. For all the runs used, the
launch site was fixed along the equater with zero longitude.
This located the site along the first point of Aires, Υ .
long the $\overline{\mathcal{I}}$ unit vector in inertial space. Due to the
restriction on the geometry of the intercept orbit (it must
be elliptical), many of the target orbits considered
yielded too little information to be examined for definite
conclusions. This was done to the fact that there were very
few elliptical intercept paths that could be flown against
them. Thus, in this section only two types of target orbits
were considered. A direct polar target orbit which passes
perpendicular to the site was used along with a moderatly
inclined orbit which also passes perpendicular to the site
(see Fig 7), (see Table I). Initially, cross-track error
data was obtained by varrying time-of-flight for the inter-
ceptor and up-dating the satellite's position from epoch by
that time-of-flight. This technique gave the range of
elliptical intercept orbits over a segment of the satellite's
orbital path, the eccentricity of each target orbit was
set at zero (circular orbit) and .2 (moderatly eccentric)
to obtain data sets for comparison. Initial covariance
values at burnout were found by inputting sigmas of 10 meter
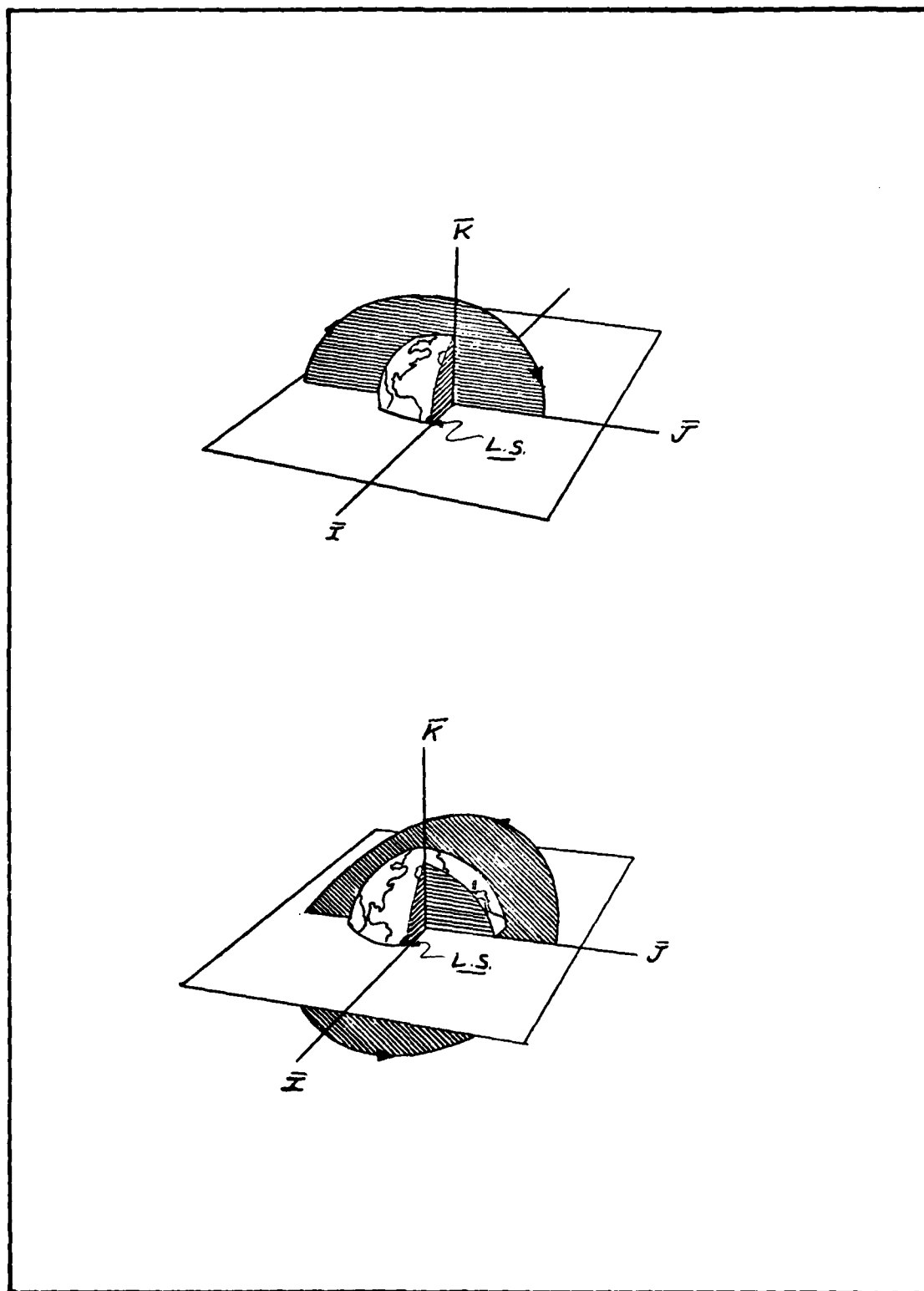
Figure 7. Polar (A) and Inclined (B) Target Orbits.

Table I: Target Orbits Examined.

Target Orbit # 1:

$A = 1.071615$
$\ell = 1 \times 10^{-8}$
$i = 90°$
$\Omega = 270°$
$\omega = 0°$
$\nu = 0°$

Target Orbit # 2:

$A = 1.071615$
$\ell = 0.2$
$i = 90°$
$\Omega = 270°$
$\omega = 0°$
$\nu = 0°$

Target Orbit # 3:

$A = 1.071615$
$\ell = 1 \times 10^{-8}$
$i = 65°$
$\Omega = 90°$
$\omega = 0°$
$\nu = 270°$

Target Orbit # 4:

$A = 1.071615$
$\ell = 0.2$
$i = 65°$
$\Omega = 90°$
$\omega = 0°$
$\nu = 270°$

in position error and .03048 meters per second (./ foot
per second) in velocity error. Then two other data sets
were developed by first zeroing out the position error at
burnout and then the velocity error. This was done to
examine the sensitivity of the resultant intercept error to
burnout position and velocity errors. This cross-track error
data for each target orbit is summarized in table II.
Plots of this data, cross-track error vs. time-of-flight,
can be seen in Figures (8 to 24). The parameter "circular-
ized cross-track error" was developed to combine the two
axis errors, as the other plots show both error ellipsoid
axes plotted as they propagated vs. time. It was found by
determining the radius of a circle with the same area as
the ellipse represented by the two axis lengths of the
cross-track slice of the error ellipsoid at the intercept
point. Next, identical data computation scheme was used
to examine error propagation when just one point on the
target orbit was used as the intercept point. Here, the
satellite's position at epoch was determined and held con-
stant as a family of elliptical intercept orbits were com-
puted for various times-of-flight. These intercept orbits
were used in the error determination to examine the effect
of orbit geometry on the intercept error growth. The
corresponding data range is shown in table III.

Table II: Intercept Errors Over Range of Target Positions.

Target # 1.

Booster Burnout Errors: Position $10\ M$

         Velocity $0.03048\ M/SEC$

Variance in the Principal Axes Lengths (M): $21.73 - 46.55$
$45.01 - 120.59$

Variance in the Circular Error Parameter (M): $31.27 - 74.92$

— — — — —

Booster Burnout Errors: Position $1 \times 10^{-3}\ M$

         Velocity $0.03048\ M/SEC$

Variance in the Principal Axes Lengths (M): $21.23 - 45.80$
$39.11 - 112.84$

Variance in the Circular Error Parameter (M): $28.81 - 74.89$

— — — — —

Booster Burnout Errors: Position $10\ M$

         Velocity $1 \times 10^{-3}\ M/SEC$

Variance in the Principal Axes Lengths (M): $4.04 - 3.88$
$22.39 - 43.18$

Variance in the Circular Error Parameter (M): $9.51 - 12.94$

Target # 2.

Booster Burnout Errors: Position $10\ M$

         Velocity $0.03048\ M/SEC$

Variance in the Principal Axes Lengths (M): $20.35 - 48.46$
$39.17 - 107.92$

Variance in the Circular Error Parameter (M): $28.23 - 67.65$

— — — — —

Table II (cont.)

——— —— —— ——

Booster Burnout Errors: Position 1×10^{-8} M

Velocity 0.03048 M/sec

Variance in the Principal Axes Lengths (M): 19.62 - 47.82
28.23 - 67.65

Variance in the Circular Error Parameter (M): 25.79 - 69.71

——— —— —— ——

Booster Burnout Errors: Position 10 M

Velocity 1×10^{-8} M/sec

Variance in the Principal Axes Lengths (M): 4.76 - 3.09
19.77 - 37.01

Variance in the Circular Error Parameter (M): 9.70 - 9.74

Target # 3.

Booster Burnout Errors: Position 10 M
Velocity 0.03048 M/sec

Variance in the Principal Axes Lengths (M): 20.72 - 40.46
26.43 - 152.84

Variance in the Circular Error Parameter (M): 23.41 - 78.63

——— —— —— ——

Booster Burnout Errors: Position 1×10^{-8} M

Velocity 0.03048 M/sec

Variance in the Principal Axes Lengths (M): 19.49 - 39.76
23.01 - 140.88

Varinace in the Circular Error Parameter (M): 21.18 - 79.84

——— —— —— ——

<u>Table II (cont.)</u>

— — — — —

Booster Burnout Errors: Position $10 M$

                           Velocity 1×10^{-8} M/SEC

Variance in the Principal Axes Lengths (M):   6.44 – 9.41
                                                        13.28 – 59.57

Variance in the Circular Error Parameter (M): 9.29 – 16.22

<u>Target # 4.</u>

Booster Burnout Errors: Position $10 M$

                           Velocity   0.03048 M/SEC

Variance in the Principal Axes Lengths (M): 18.69 – 40.18
                                                          24.42 – 138.08

Variance in the Circular Error Parameter (M): 21.36 – 74.49

— — — — —

Booster Burnout Errors: Position 1×10^{-8} M

                           Velocity   0.03048 M/SEC

Varinace in the Principal Axes Lengths (M): 17.21 – 39.52
                                                          20.75 – 127.09

Variance in the Circular Error Parameter (M): 18.89 – 70.87

— — — — —

Booster Burnout Errors: Position $10 M$

                           Velocity 1×10^{-8} M/SEC

Variance in the Principal Axes Lengths (M): 6.75 – 4.22
                                                         13.17 – 59.31

Variance in the Circular Error Parameter (M): 9.43 – 15.15

— — — — —

Table # III: Intercept Errors for a Single Family of Intercept Orbits.

Target # 1.

Booster Burnout Errors: Position $10 M$

　　　　　　　　　　　　Velocity 0.03048 M/sec

Variance in the Principal Axes Lengths (M): $22.03 - 39.55$
$44.07 - 155.77$

Variance in the Circular Error Parameter (M): $31.16 - 78.49$

— — — —

Booster Burnout Error: Position $1 \times 10^{-8} M$

　　　　　　　　　　　　Velocity 0.03048 M/sec

Variance in the Principal Axes Lengths (M): $21.44 - 38.86$
$38.46 - 144.07$

Variance in the Circular Error Parameter (M): $28.72 - 74.82$

— — — —

Booster Burnout Errors: Position $10 M$

　　　　　　　　　　　　Velocity 1×10^{-8} M/sec

Variance in the Principal Axes Lengths (M): $4.29 - 4.61$
$21.66 - 59.49$

Variance in the Circular Error Parameter (M): $9.64 - 16.57$

Target # 2.

Booster Burnout Errors: Position $10 M$

　　　　　　　　　　　　Velocity 0.03048 M/sec

Variance in the Principal Axes Lengths (M): $18.42 - 35.11$
$35.74 - 151.68$

Variance in the Circular Error Parameter (M): $25.65 - 72.98$

— — — —

Booster Burnout Errors: Position 1×10^{-8} M

Velocity   0.03048 M/sec

Variance in the Principal Axes Lengths (M):   17.54 - 34.51
31.99 - 134.47

Variance in the Circular Error Parameter (M):   23.07 - 69.38

Booster Burnout Errors: Position   10 M

Velocity 1×10^{-8} M/sec

Variance in the Principal Axes Lengths (M):   5.11 - 3.98
19.00 - 59.83

Variance in the Circular Error Parameter (M):   9.85 - 15.42

Target # 3.

Booster Burnout Errors: Position   10 M

Velocity   0.03048 M/sec

Variance in the Principal Axes Lengths (M):   19.59 - 37.93
23.48 - 140.42

Variance in the Circular Error Parameter (M):   21.45 - 72.98

Booster Burnout Errors: Position 1×10^{-8} M

Velocity   0.03048 M/sec

Variance in the Principal Axes Lengths (M):   18.13 - 37.62
20.13 - 129.02

Variance in the Circular Error Parameter (M):   19.10 - 69.67

Table # III: (cont.)

— — — —

Booster Burnout Errors: Position 10 m

Velocity 1×10^{-8} m/sec

Variance in the Principal Axes Lengths (M): 7.25 - 2.84
12.20 - 55.55

Variance in the Circular Error Parameter (M): 9.40 - 12.57

— — — —

Target # 4.

Booster Burnout Errors: Position 10 m

Velocity 0.03048 m/sec

Variance in the Principal Axes Lengths (M): 18.57 - 36.23
24.42 - 137.61

Variance in the Circular Error Parameter (M): 22.08 - 73.29

— — — —

Booster Burnout Errors: Position 1×10^{-8} m

Velocity 0.03048 m/sec

Variance in the Principal Axes Lengths (M): 18.08 - 36.05
20.72 - 129.57

Variance in the Circular Error Parameter (M): 19.35 - 68.34

— — — —

Booster Burnout Errors: Position 10 m

Velocity 1×10^{-8} m/sec

Variance in the Principal Axes Lengths (M): 7.02 - 2.11
12.75 - 54.65

Variance in the Circular Error Parameter (M): 9.96 - 10.74

— — — —

FIGURES 8 - 11.

Data for the Range of Target Positions

TARGET # 1.

Figure # 8

Figuure # 9

36

Figure # 10

37

Figure # 11

38
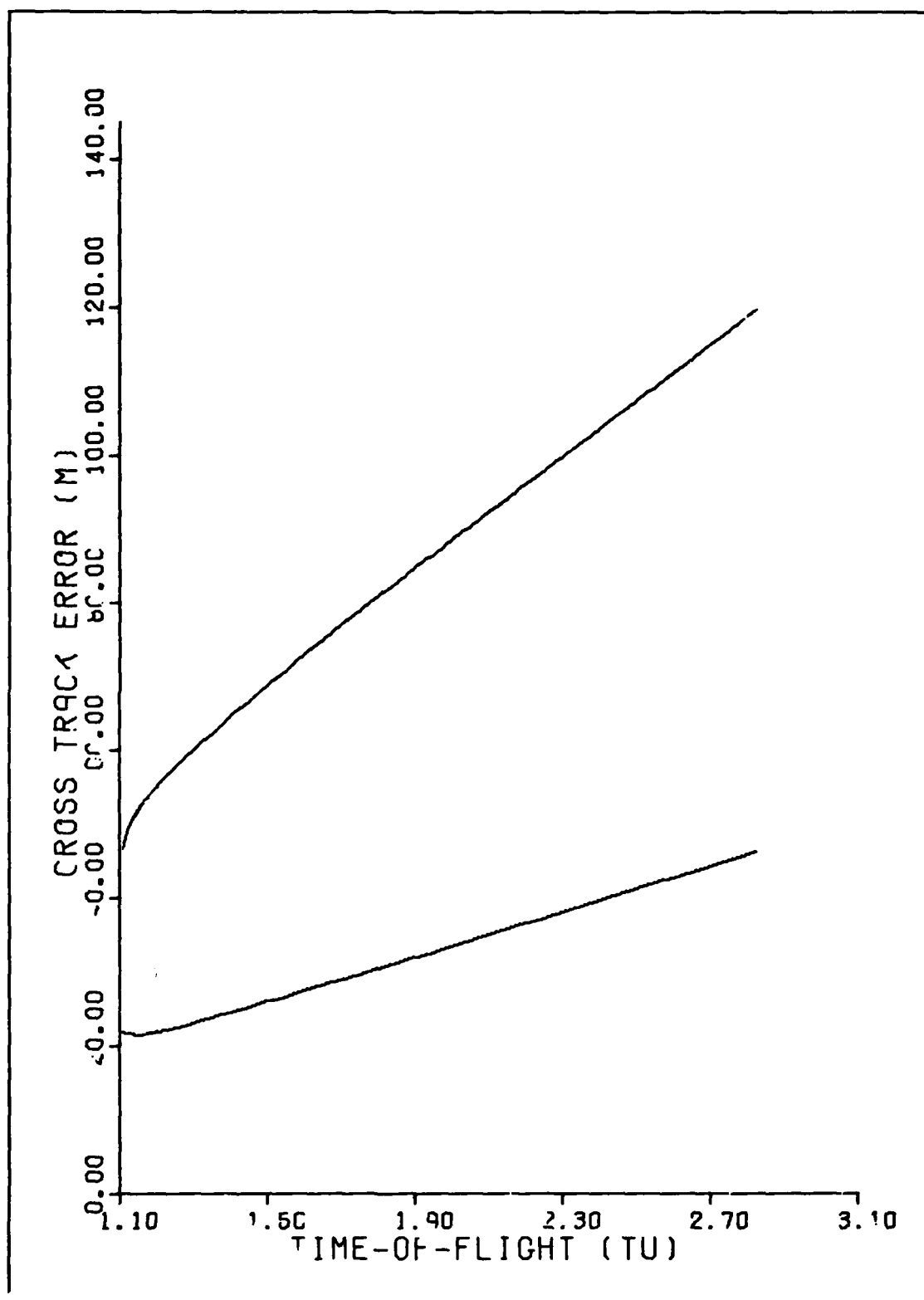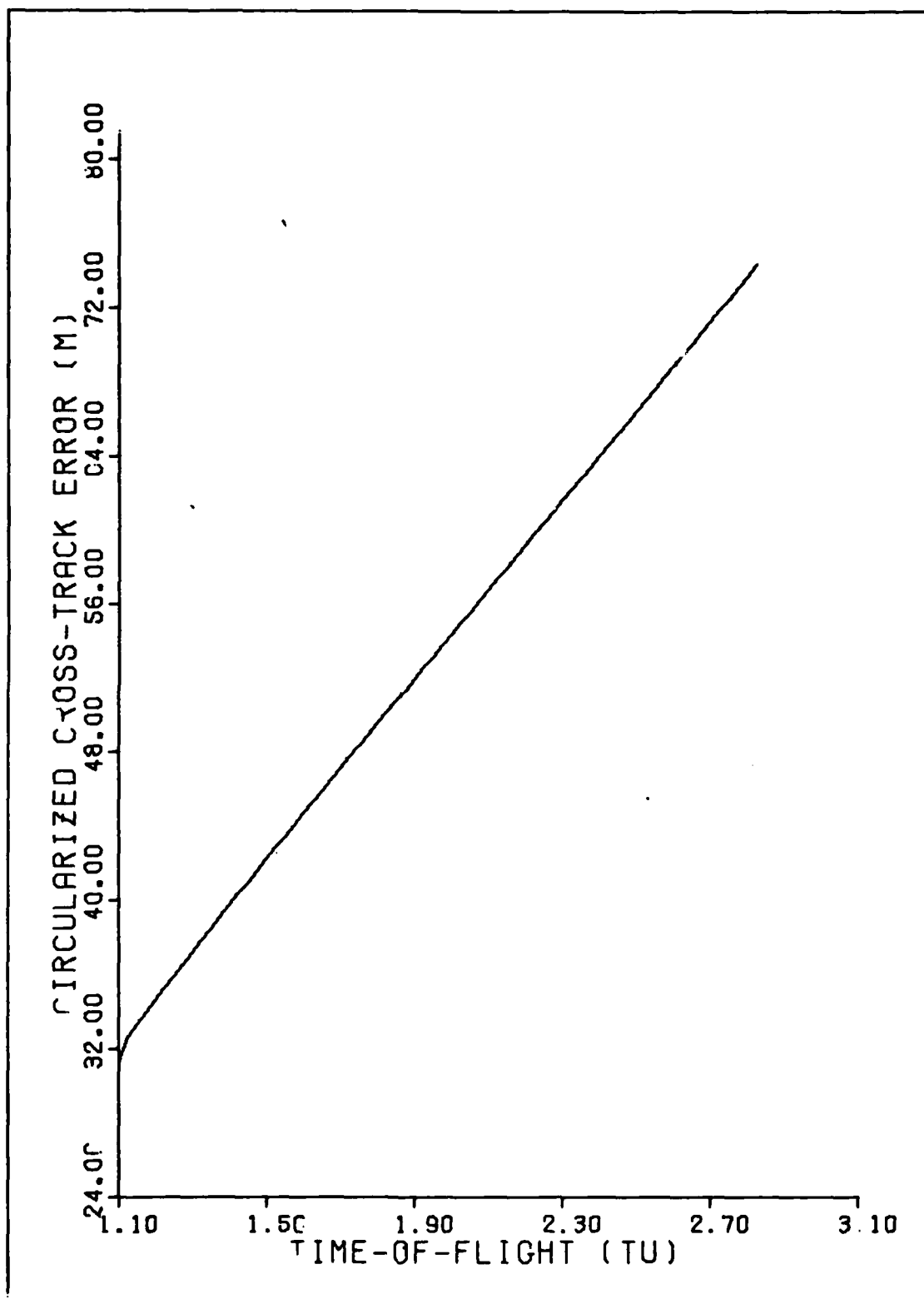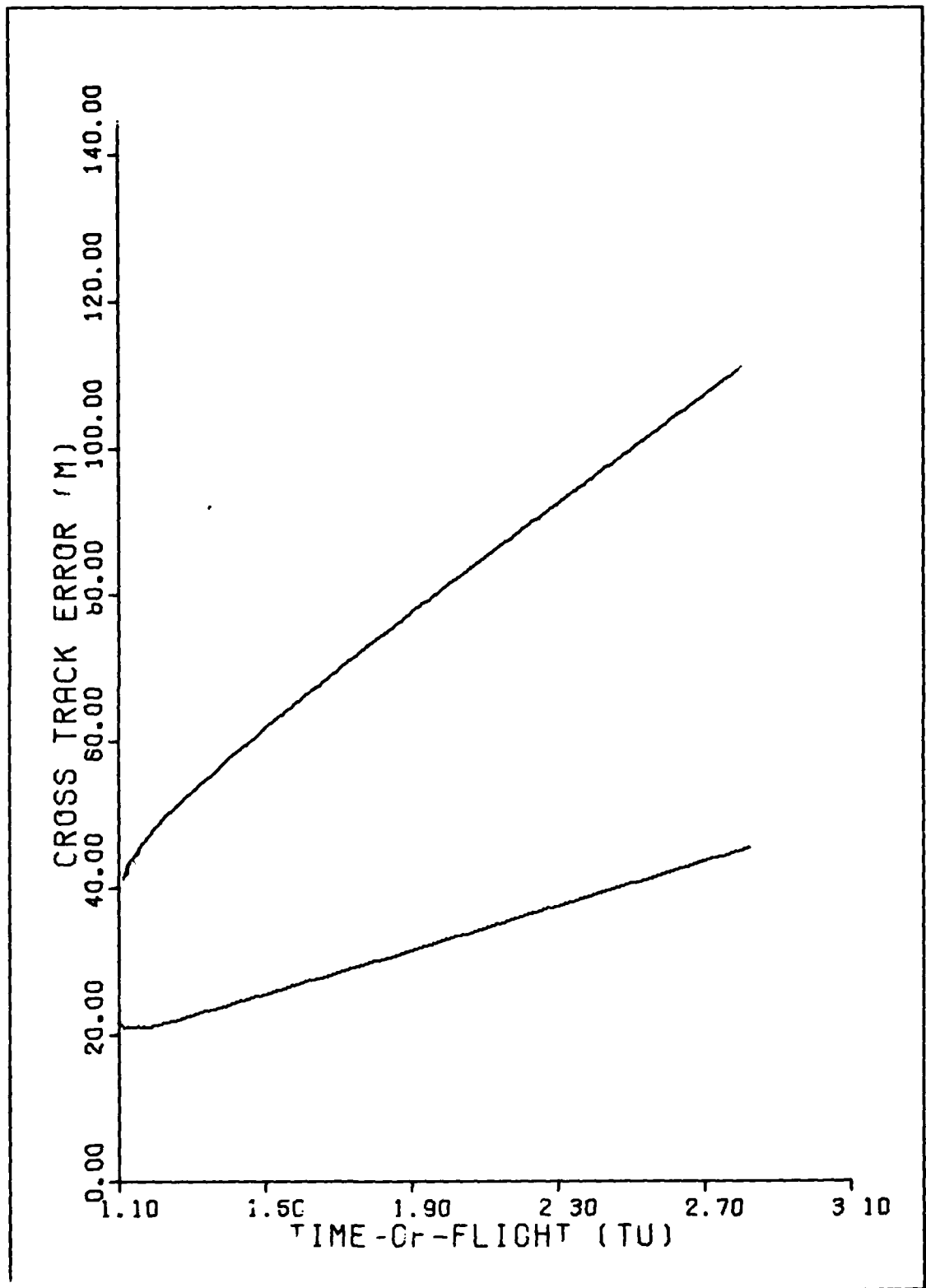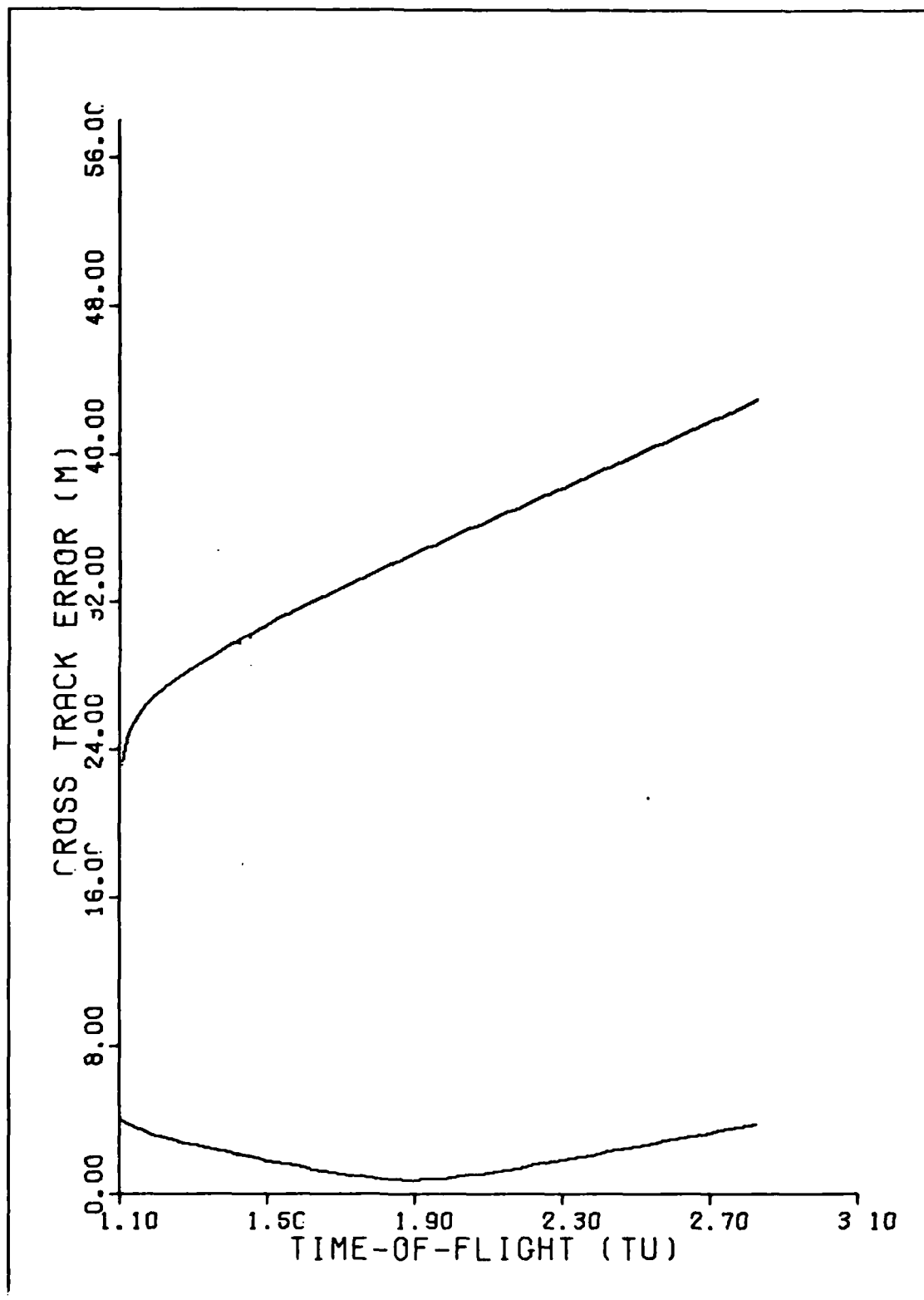
FIGURES 12-15.

Data for the Range of Target Positions

TARGET # 2.
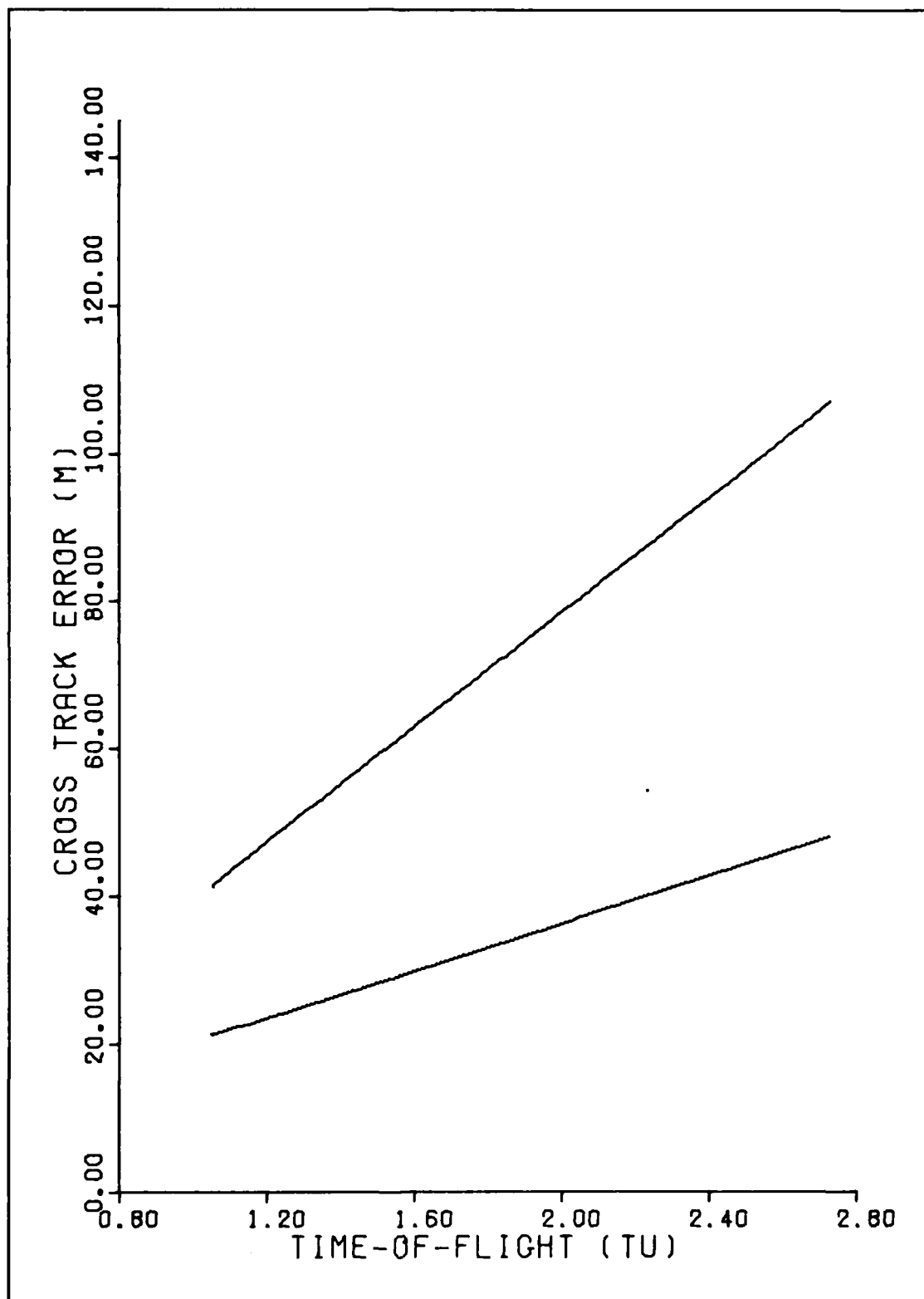
Figure # 12

Figure # 13

41

Figure # 14

42

Figure # 15

43
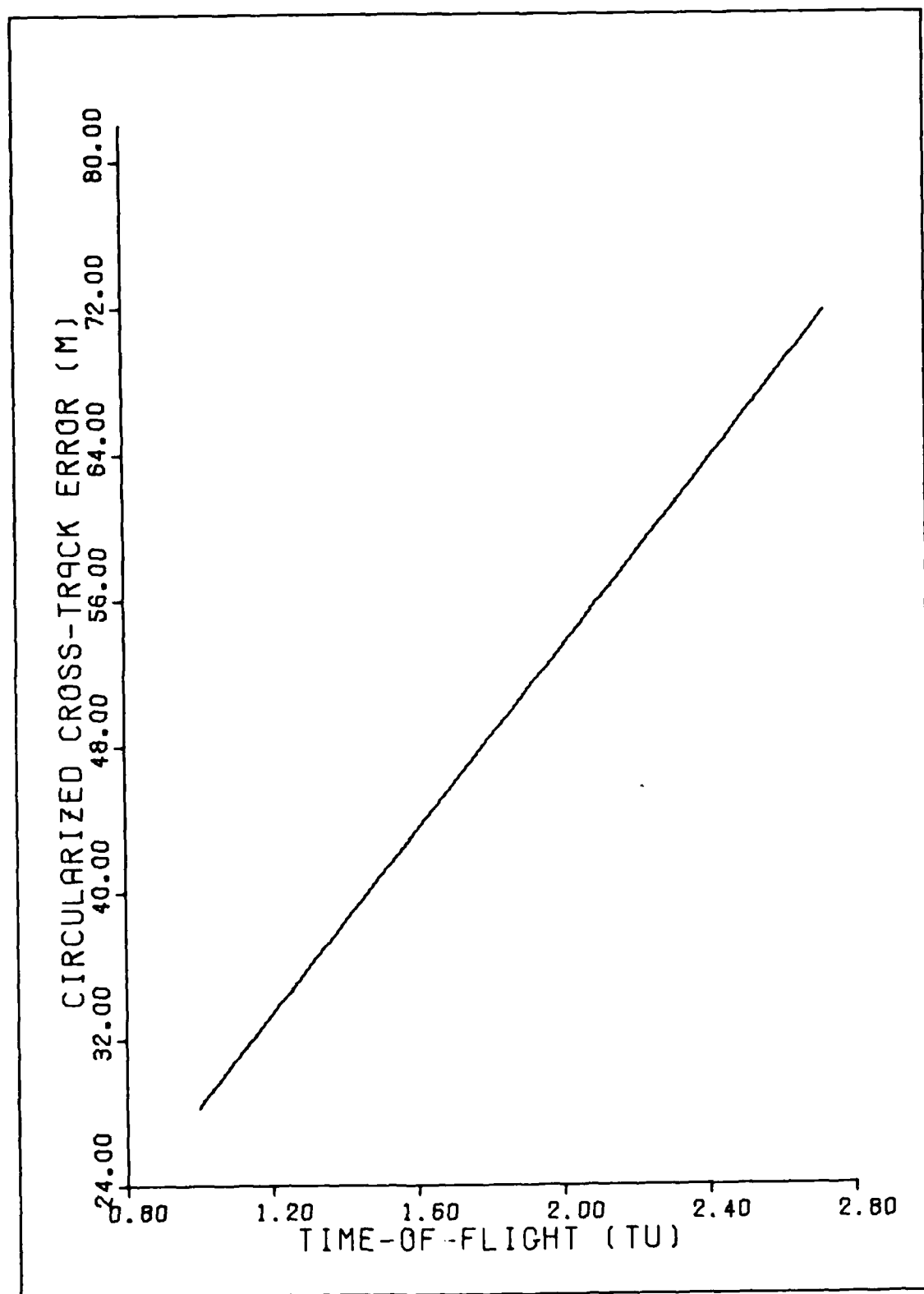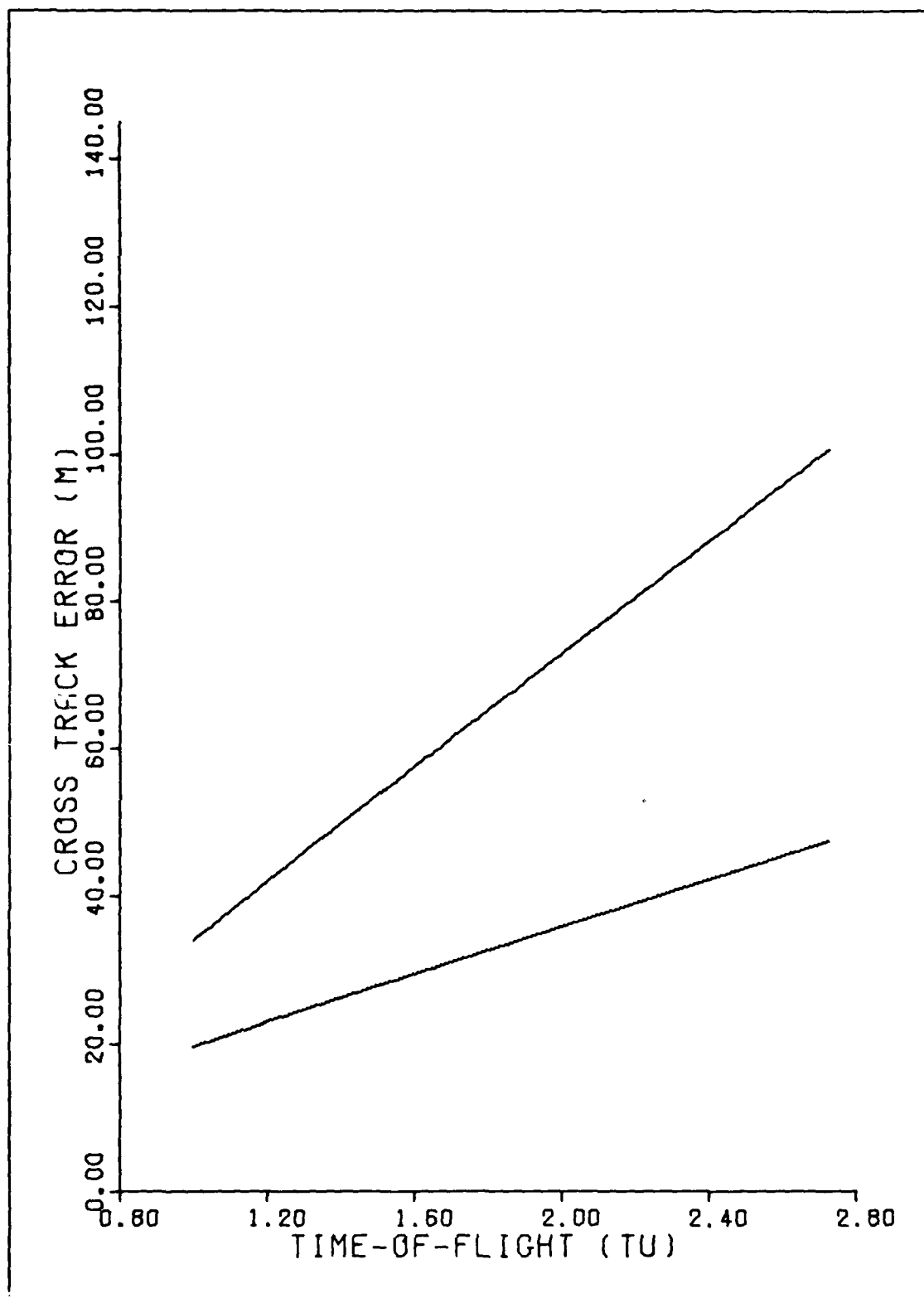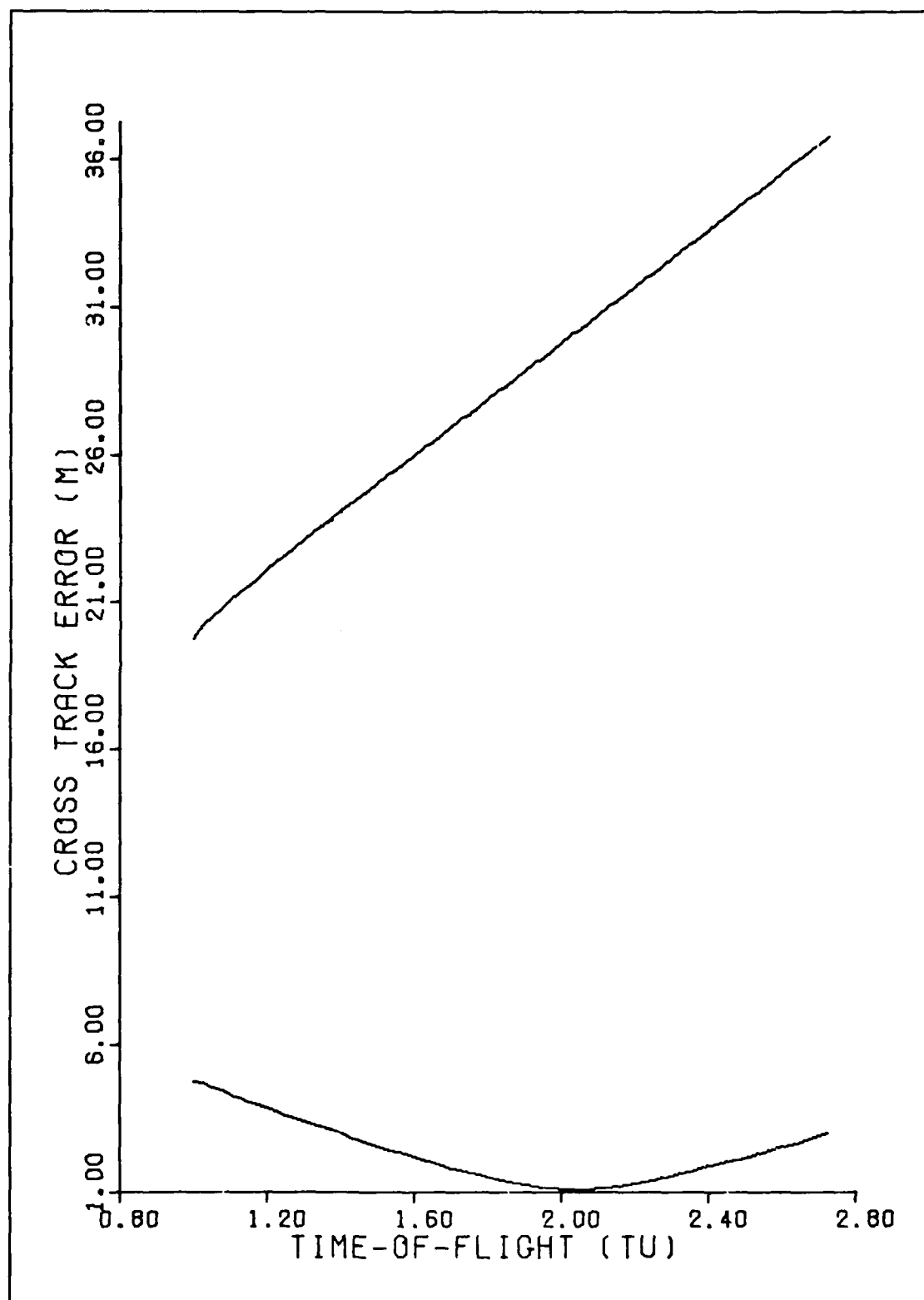
FIGURES 16 - 19.

Data for the Range of Target Positions

TARGET # 3.

44
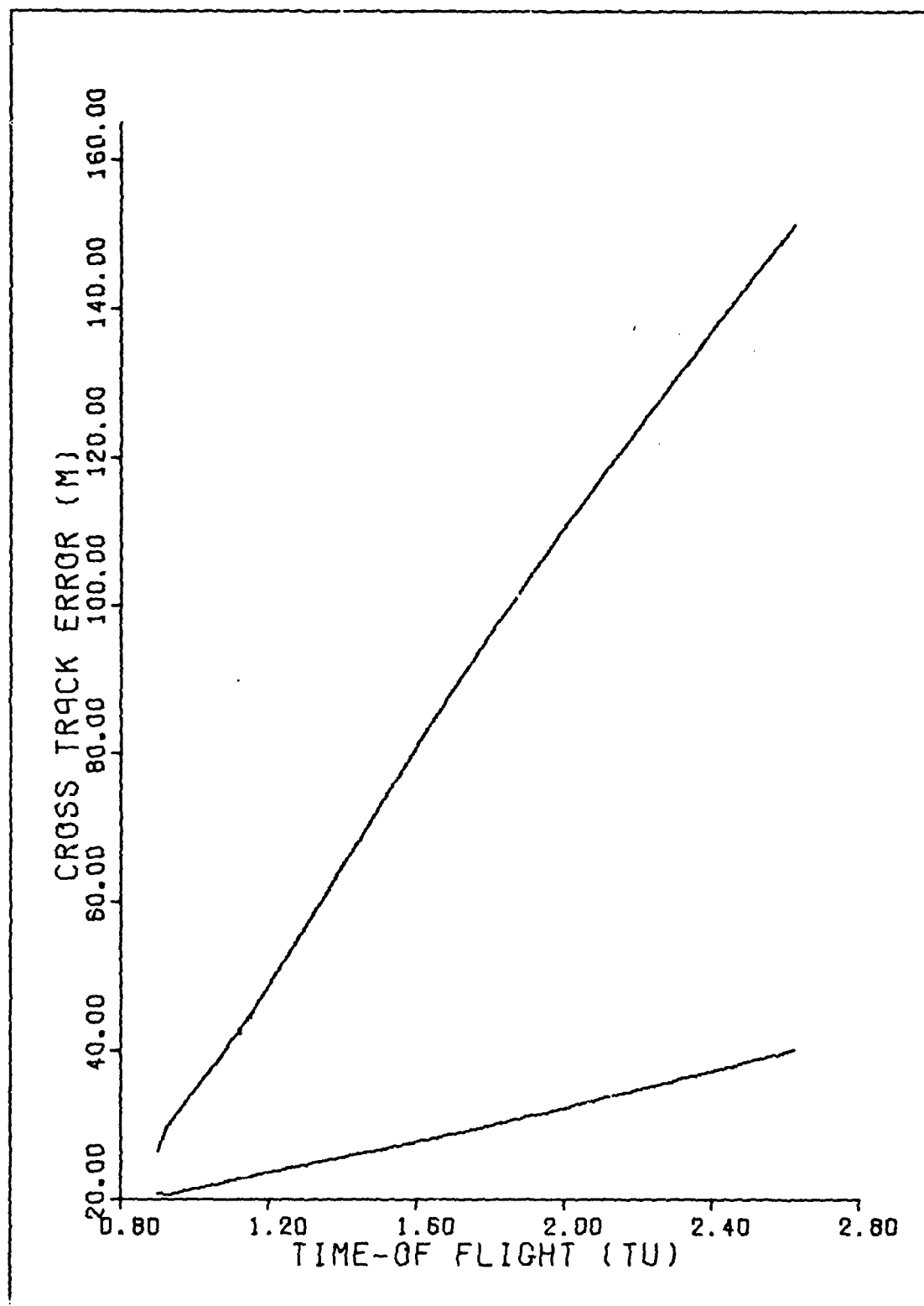
Figure # 16

Figure # 17

46

Figure #18

47

Figure # 19

48
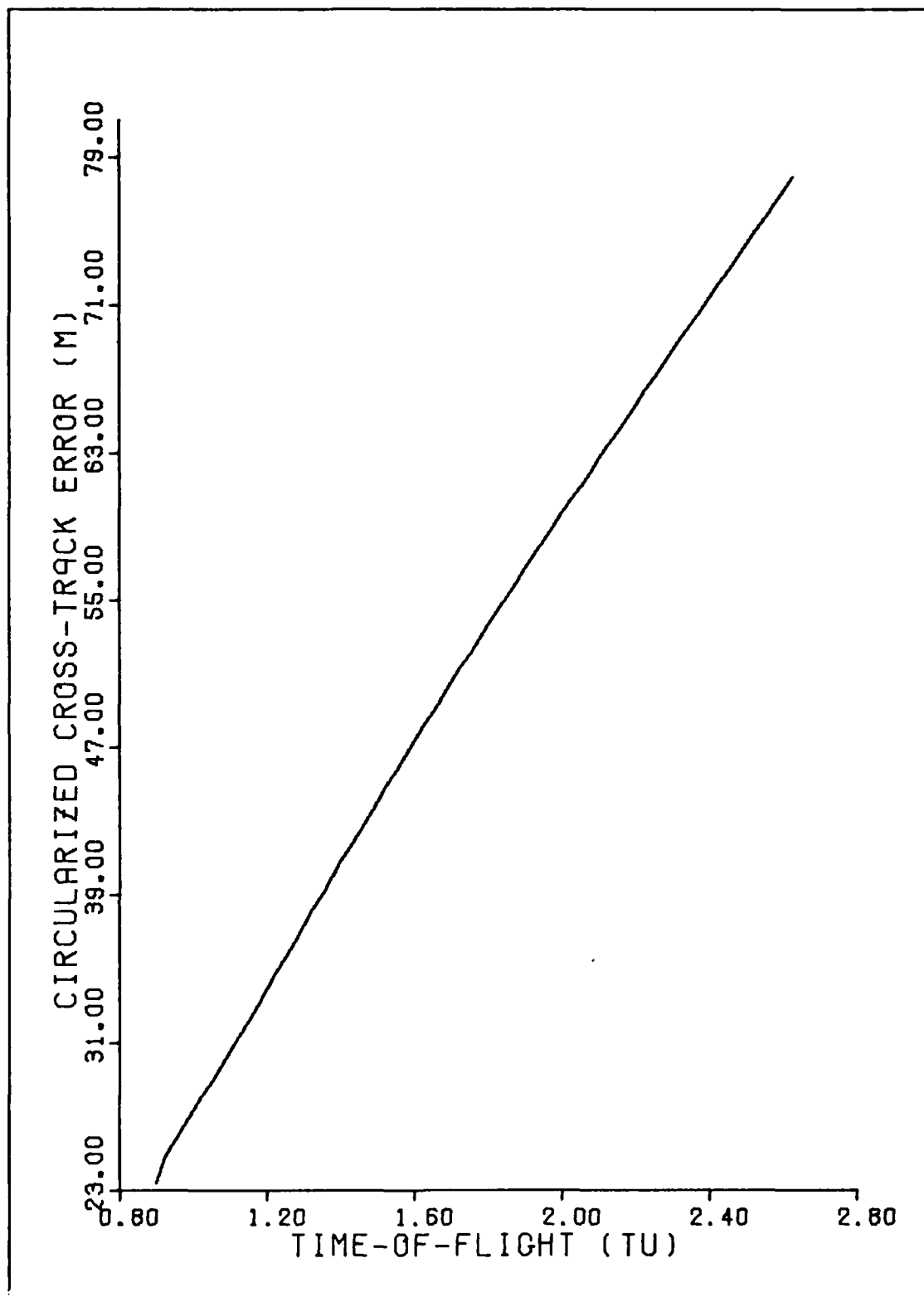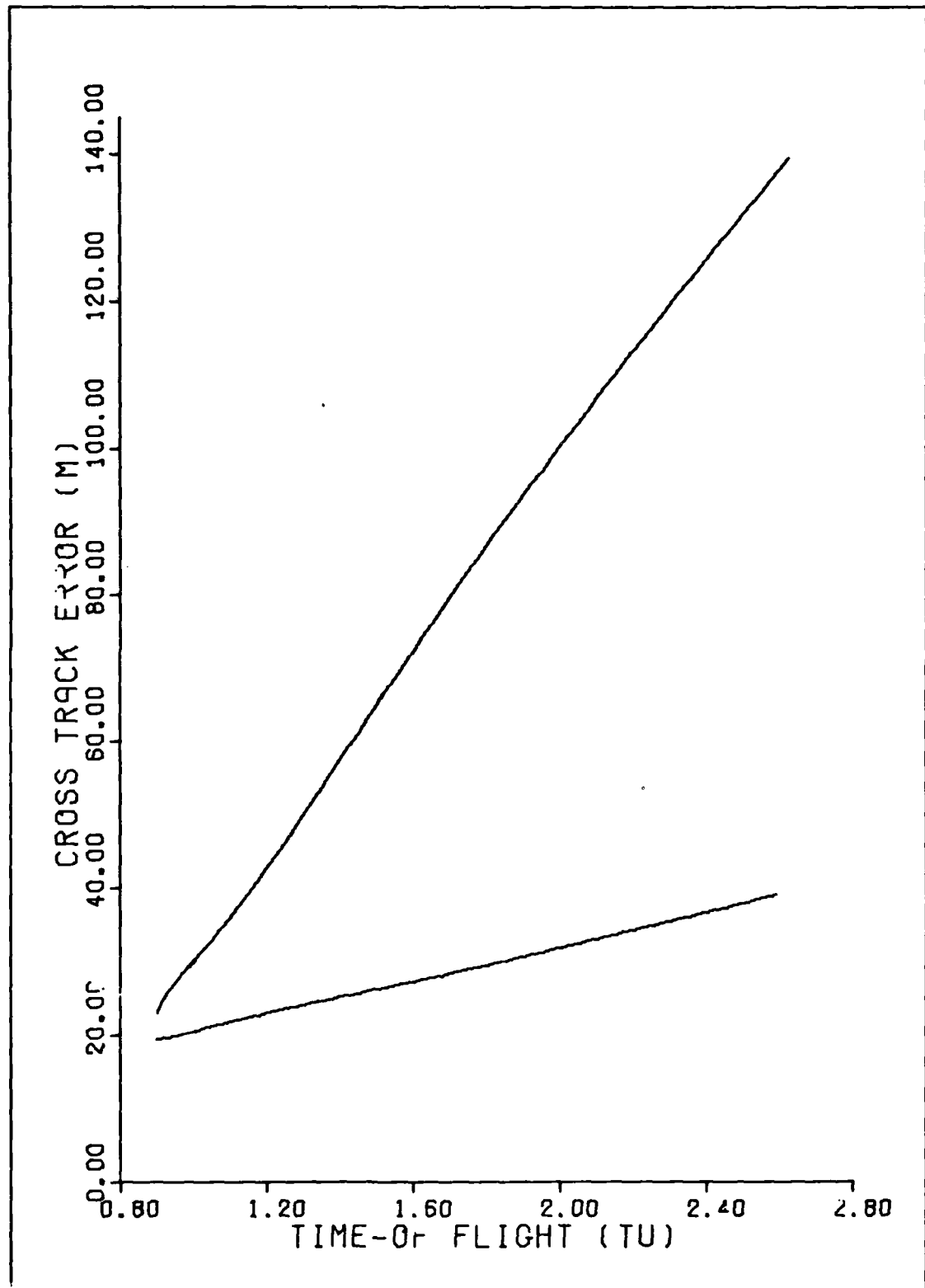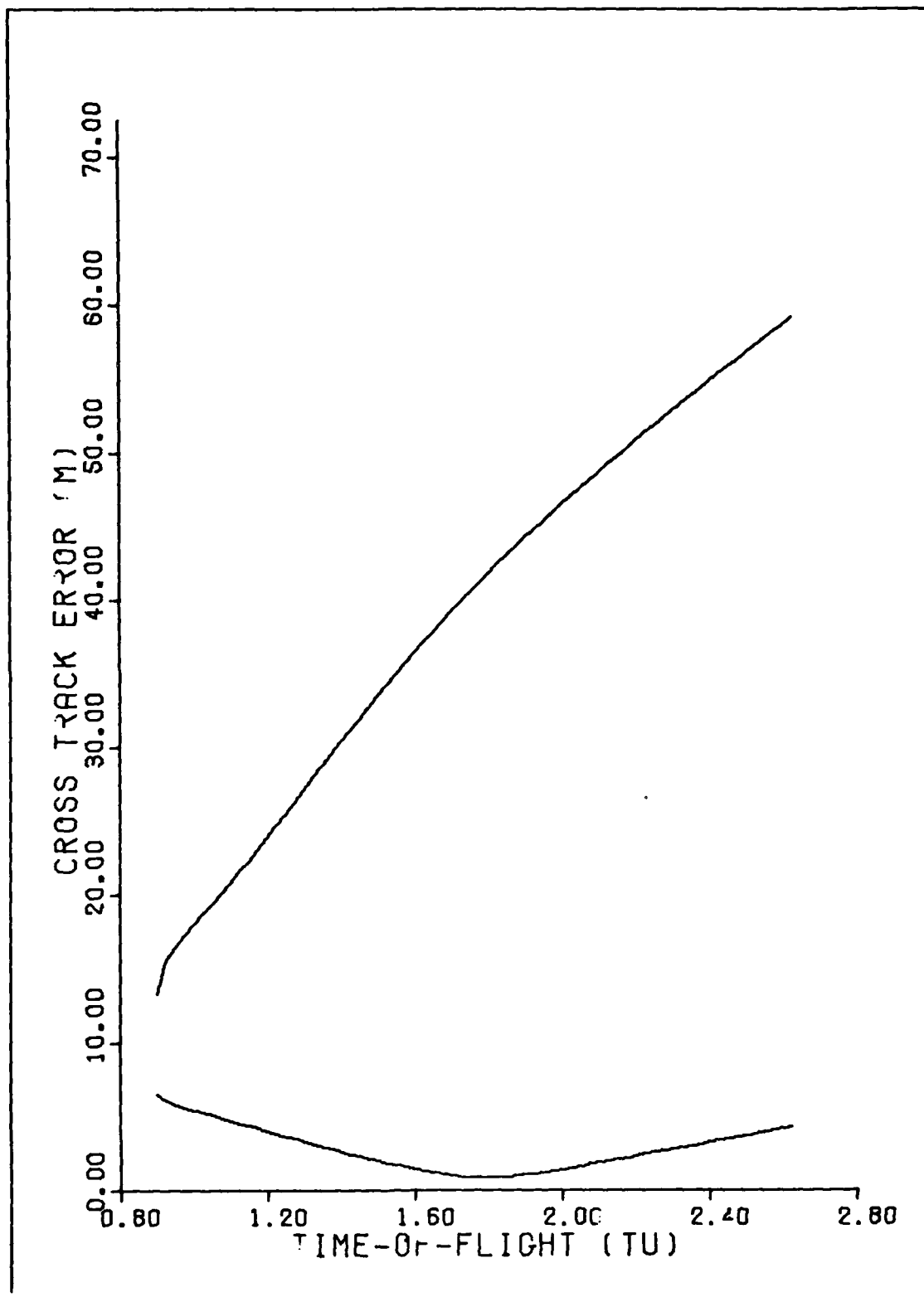
FIGURES 20 - 23.

Data for the Range of Target Positions

TARGET # 4.

Figure # 26

50
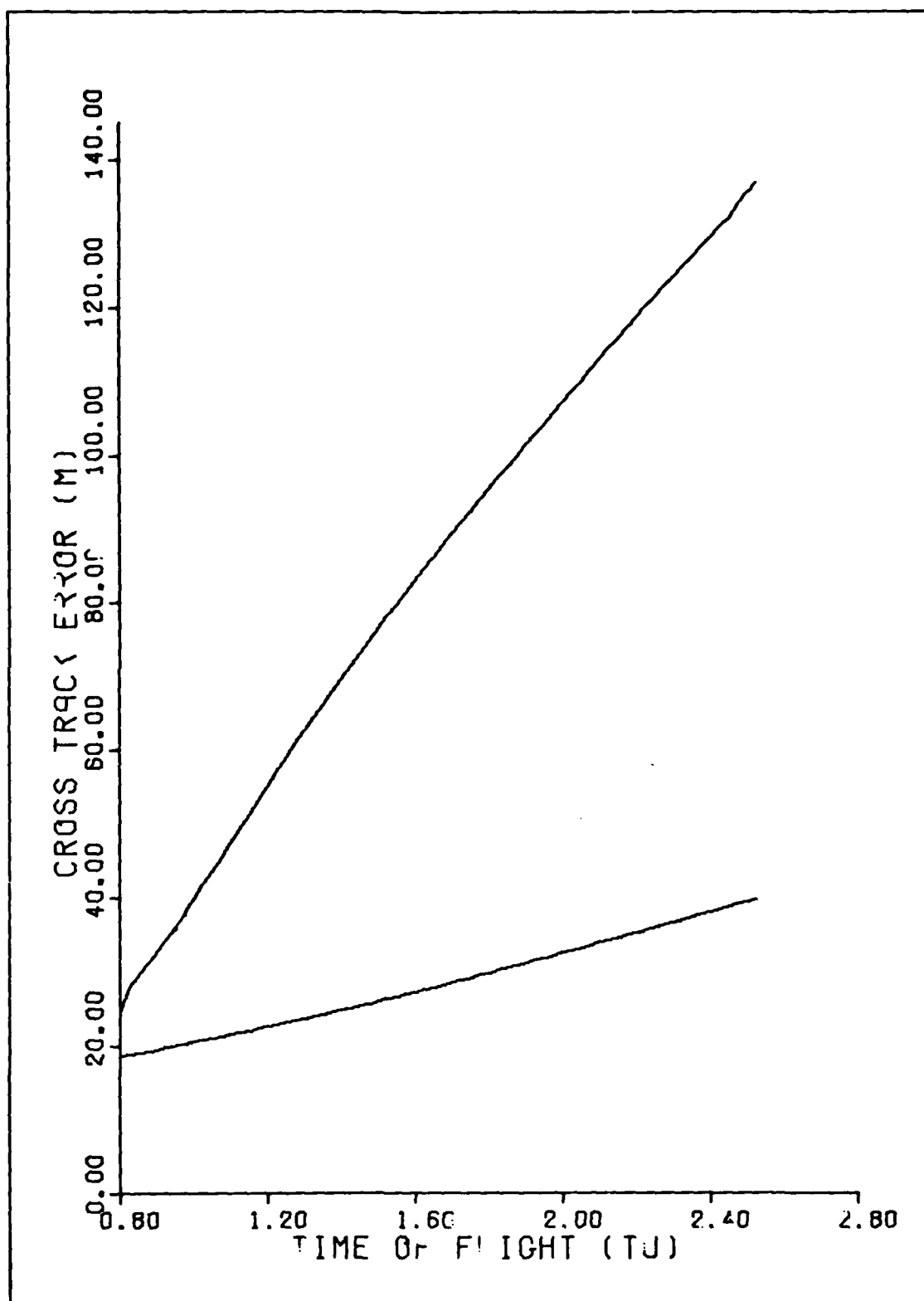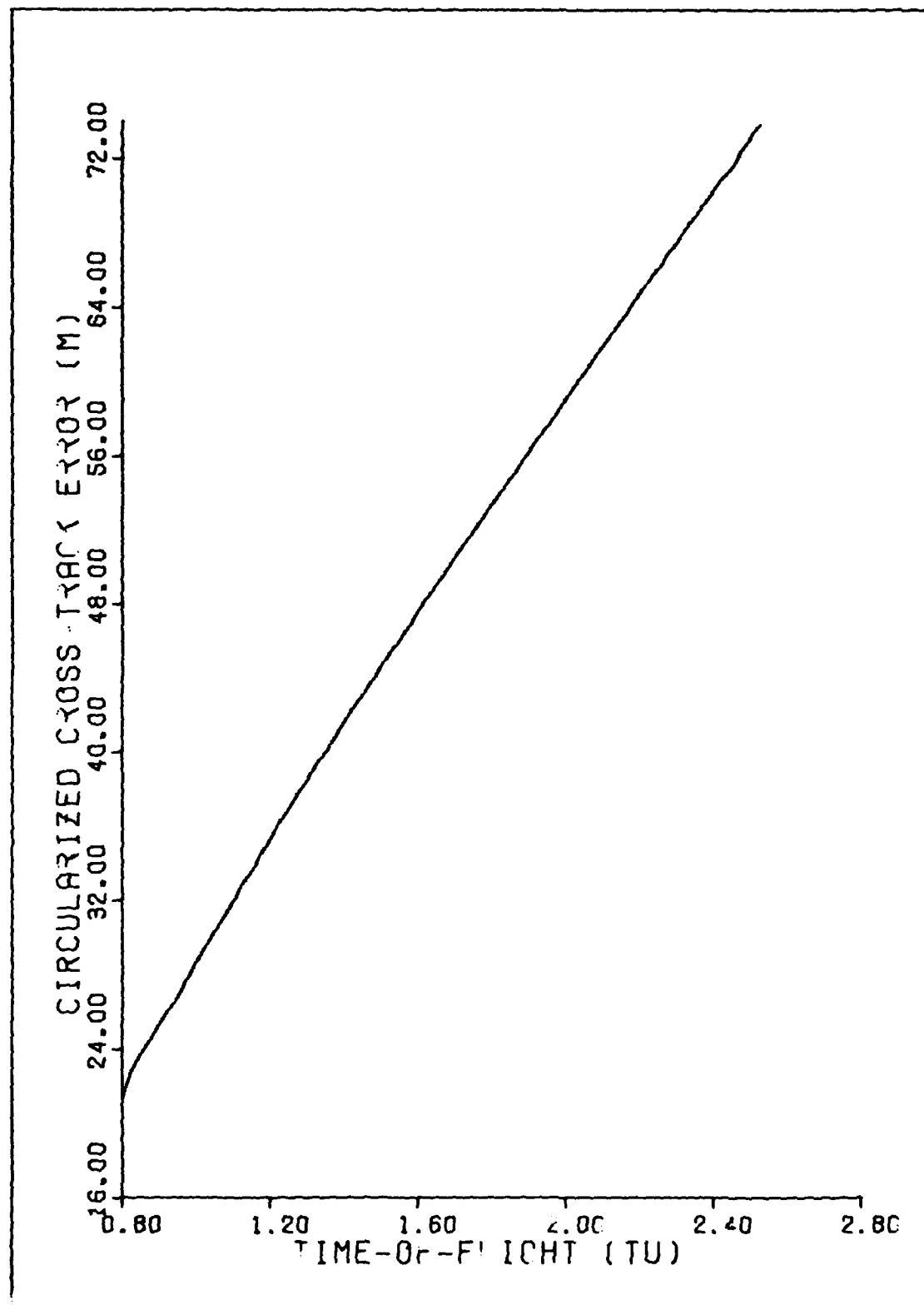
Figure # 21

Figure # 22

52
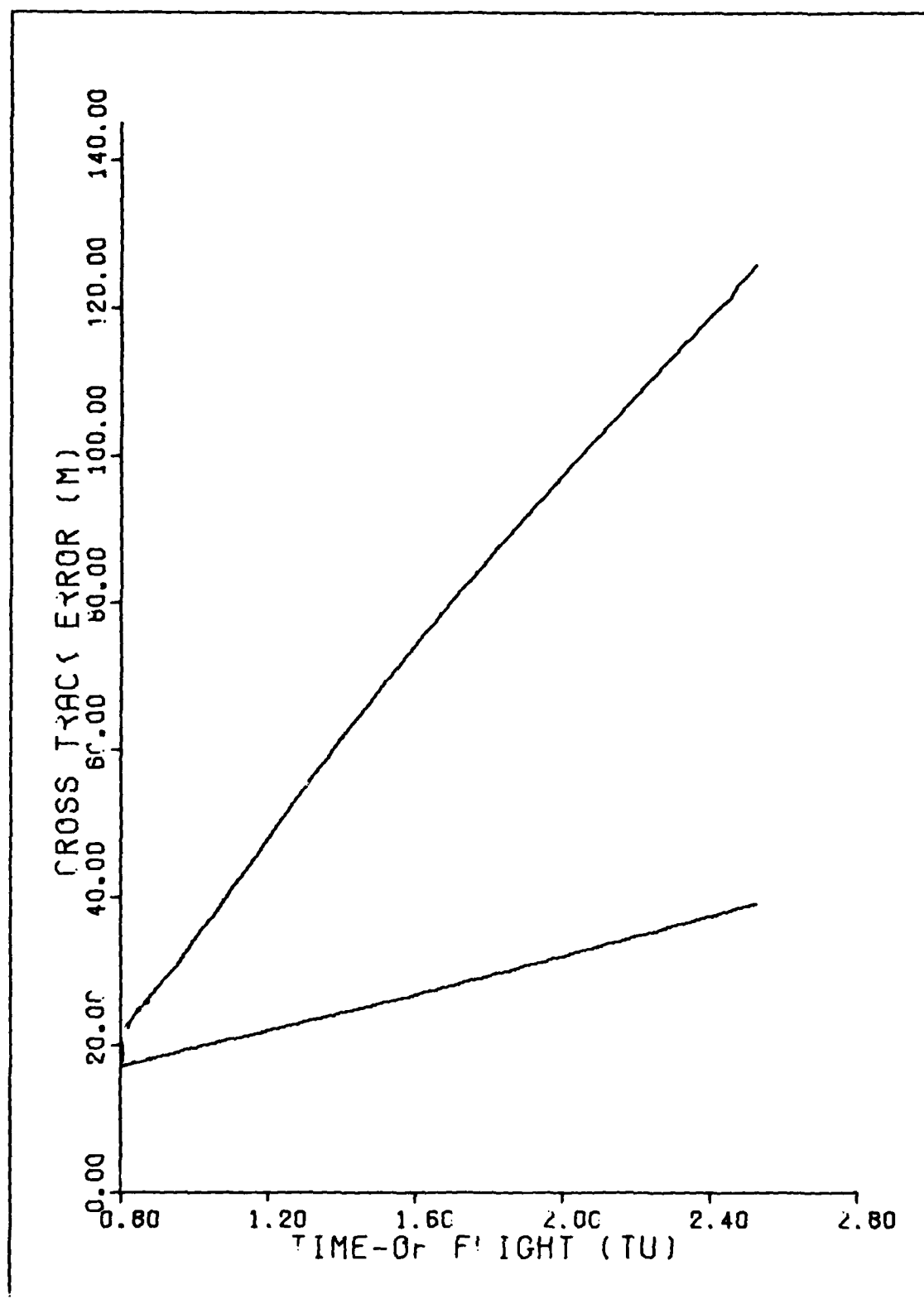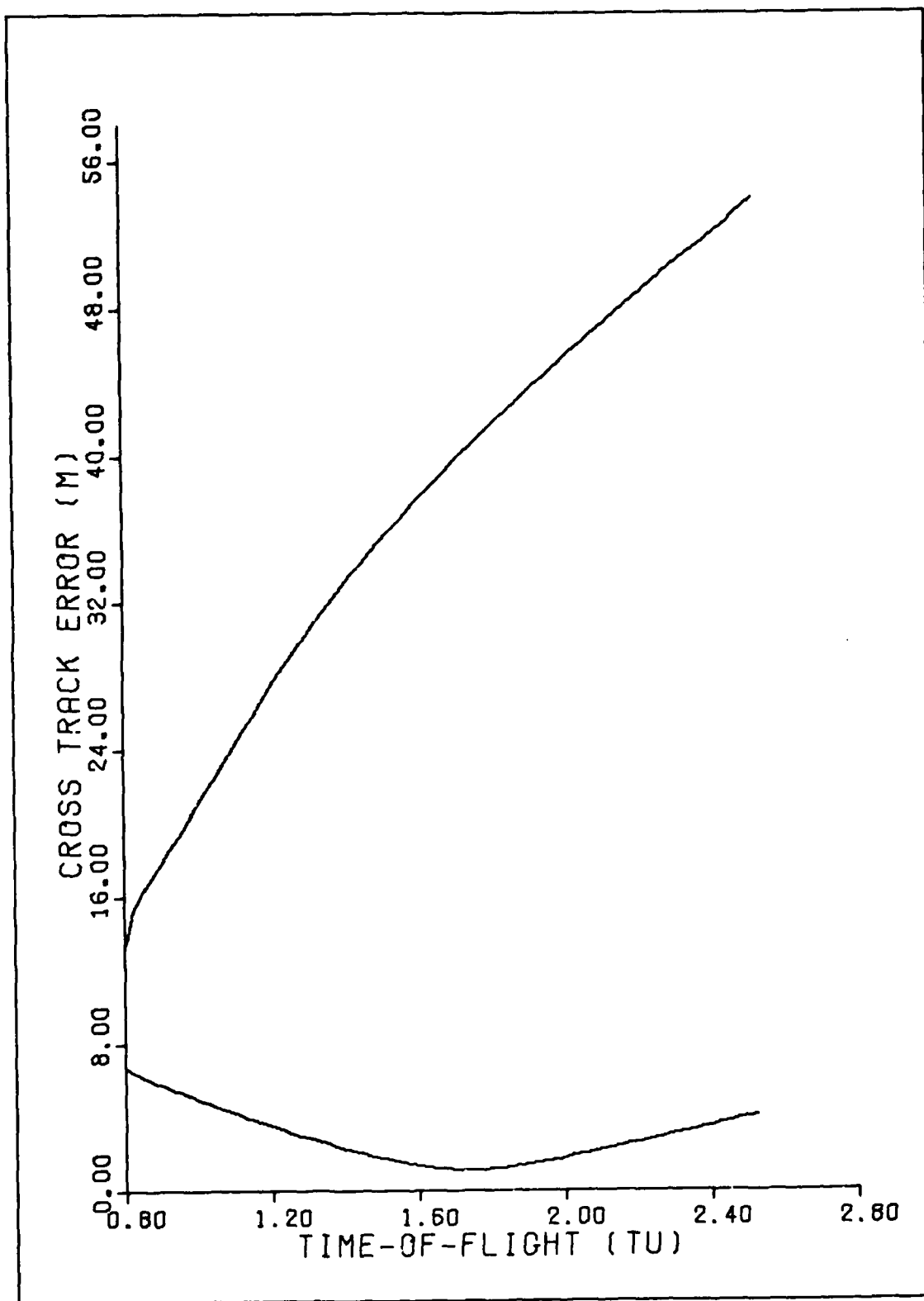
Figure # 23

53

FIGURES 24 - 27.

Single Family of Intercepts

TARGET # 1.

Figure # 24

55

Figure # 25

56
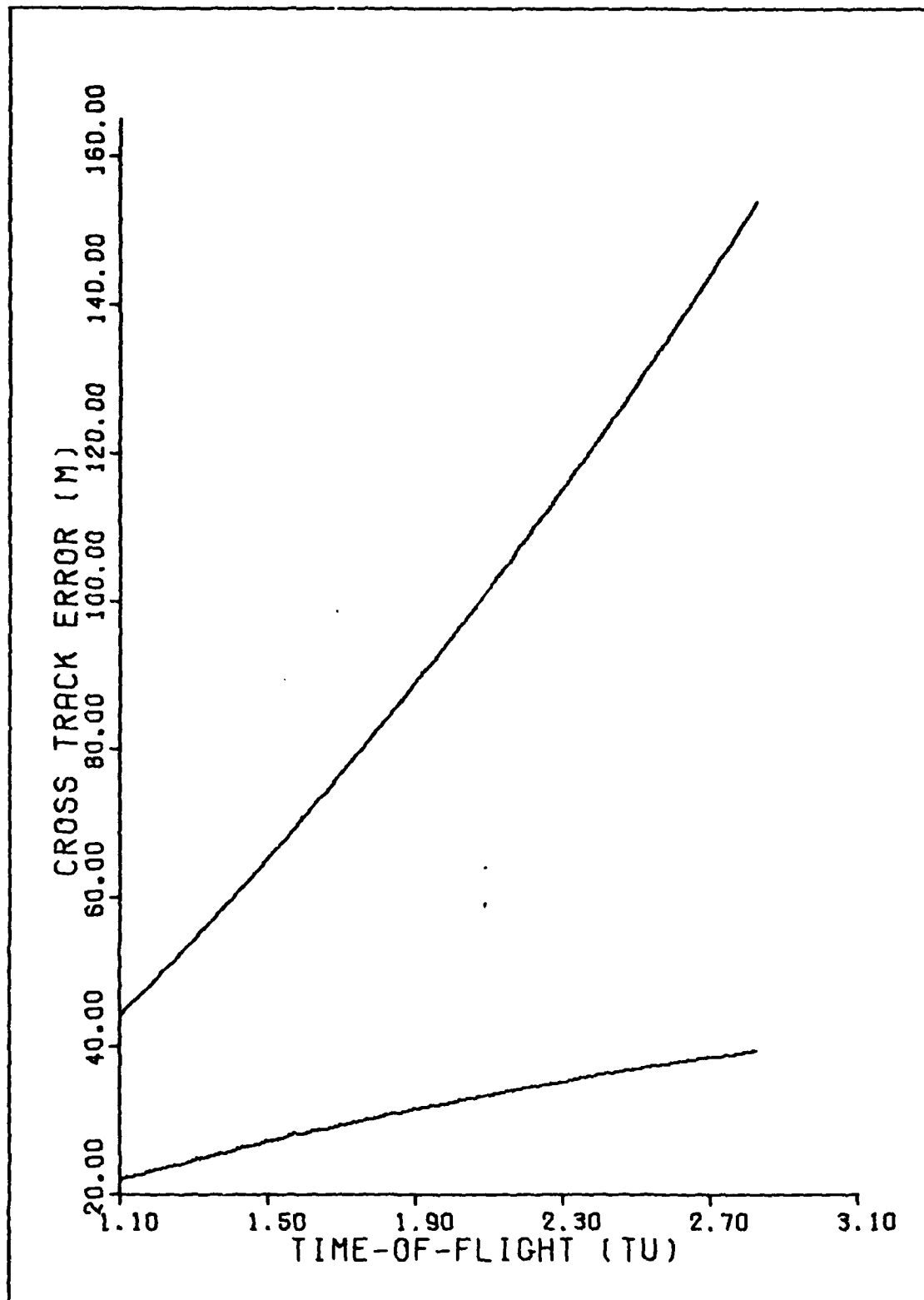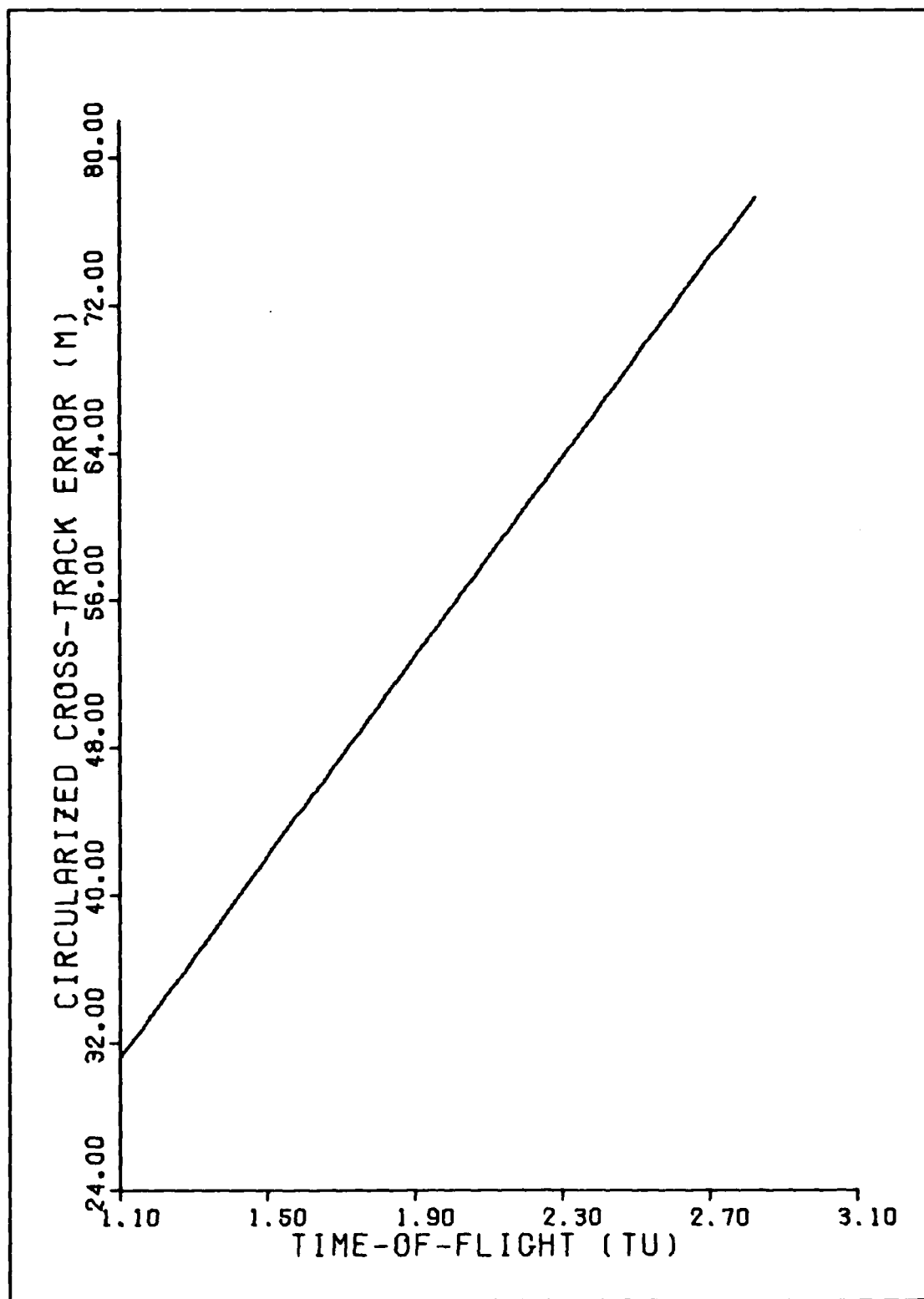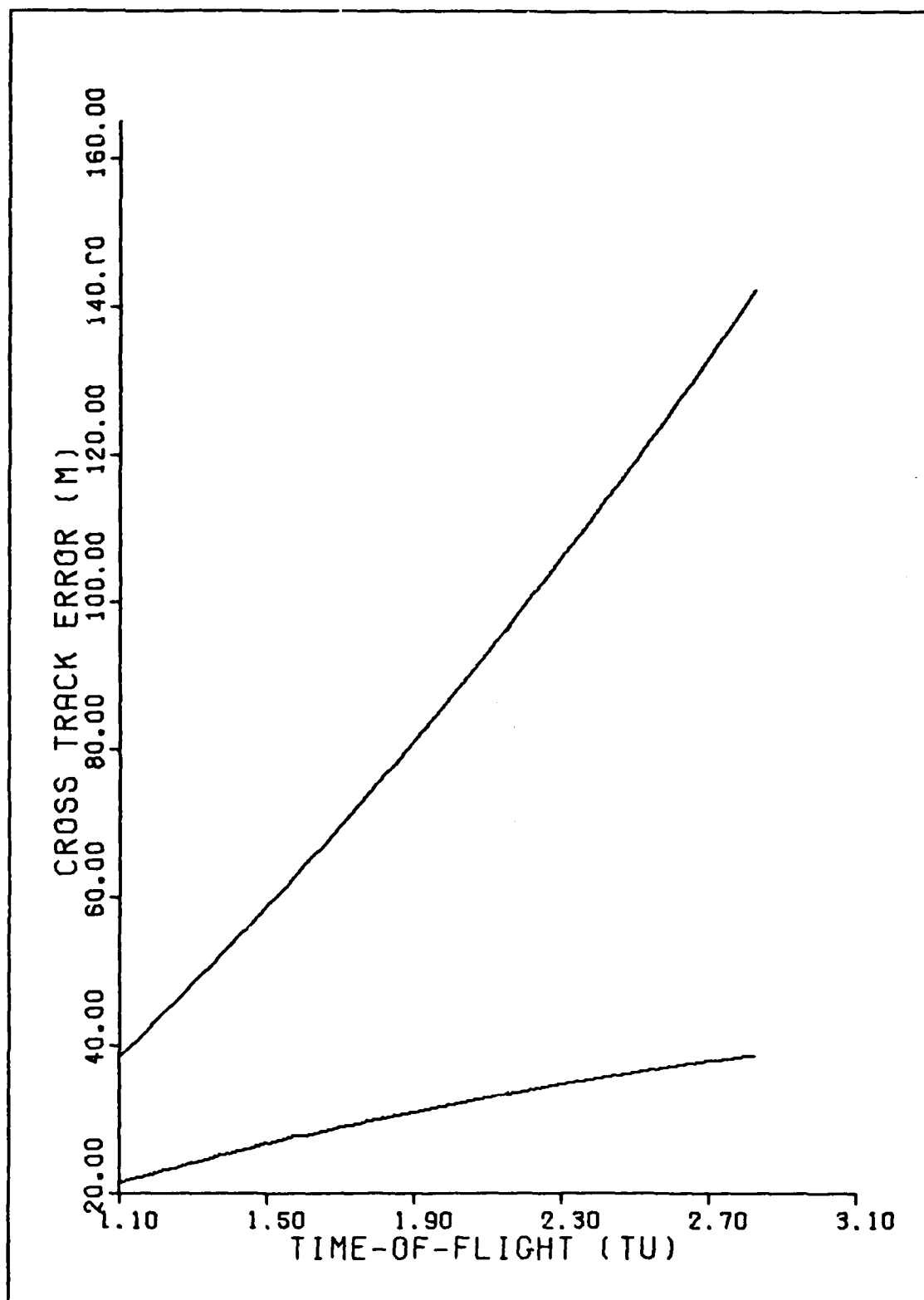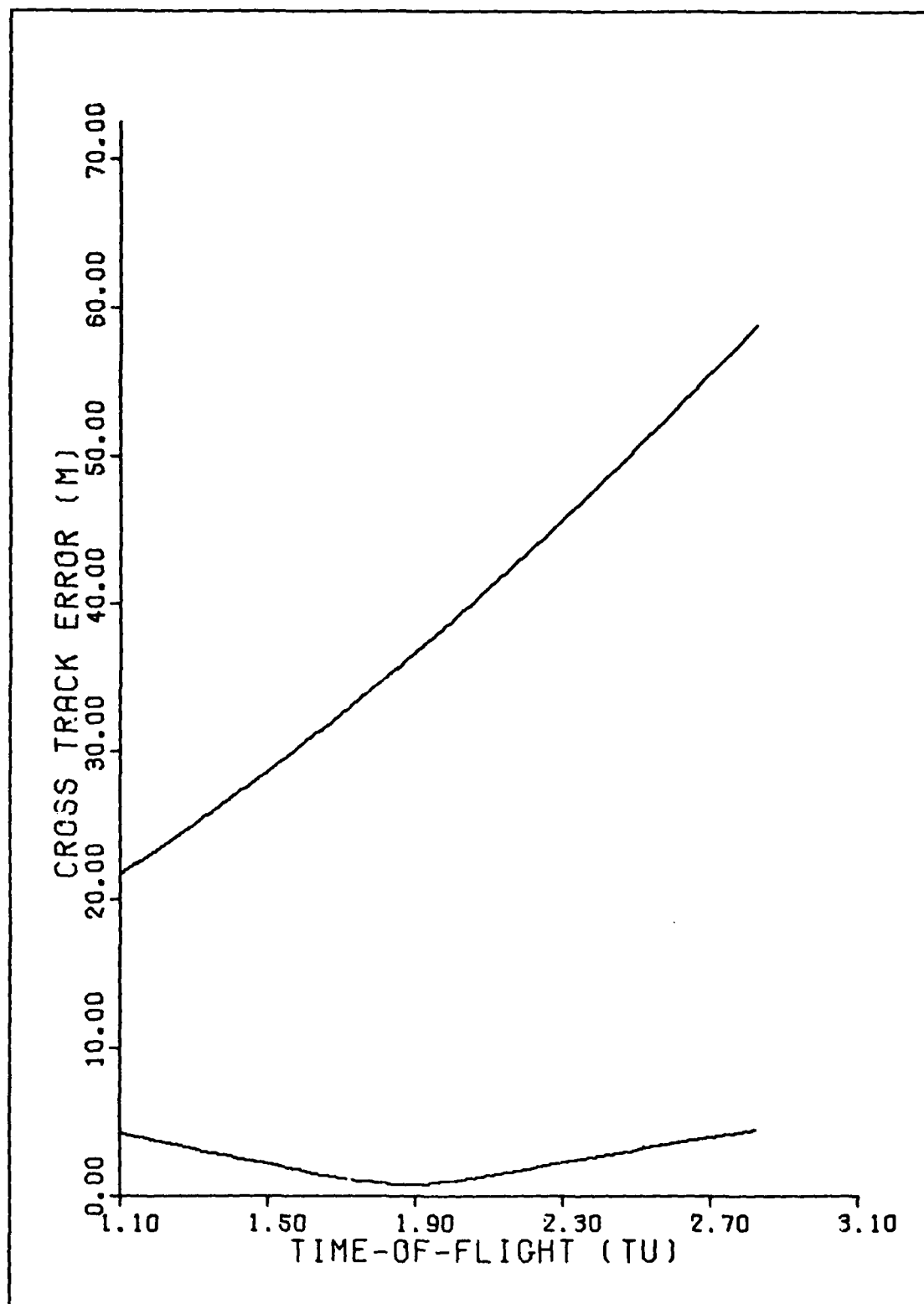
Figure # 26

Figure # 27

FIGURES 28 - 31.

Single Family of Intercepts

TARGET # 2.

Figure # 28

60

Figure # 29

61

Figure # 30

62

Figue # 31

63
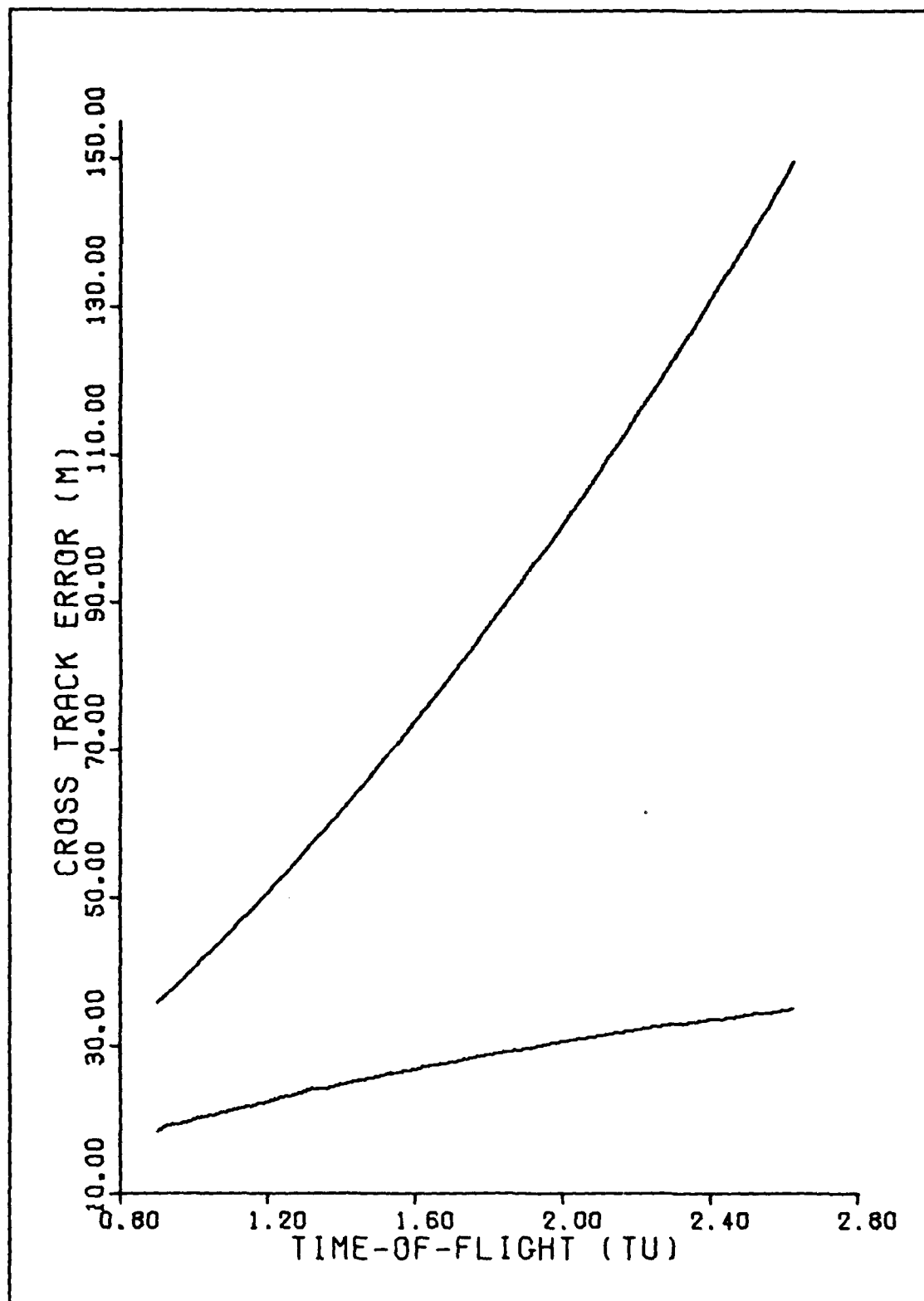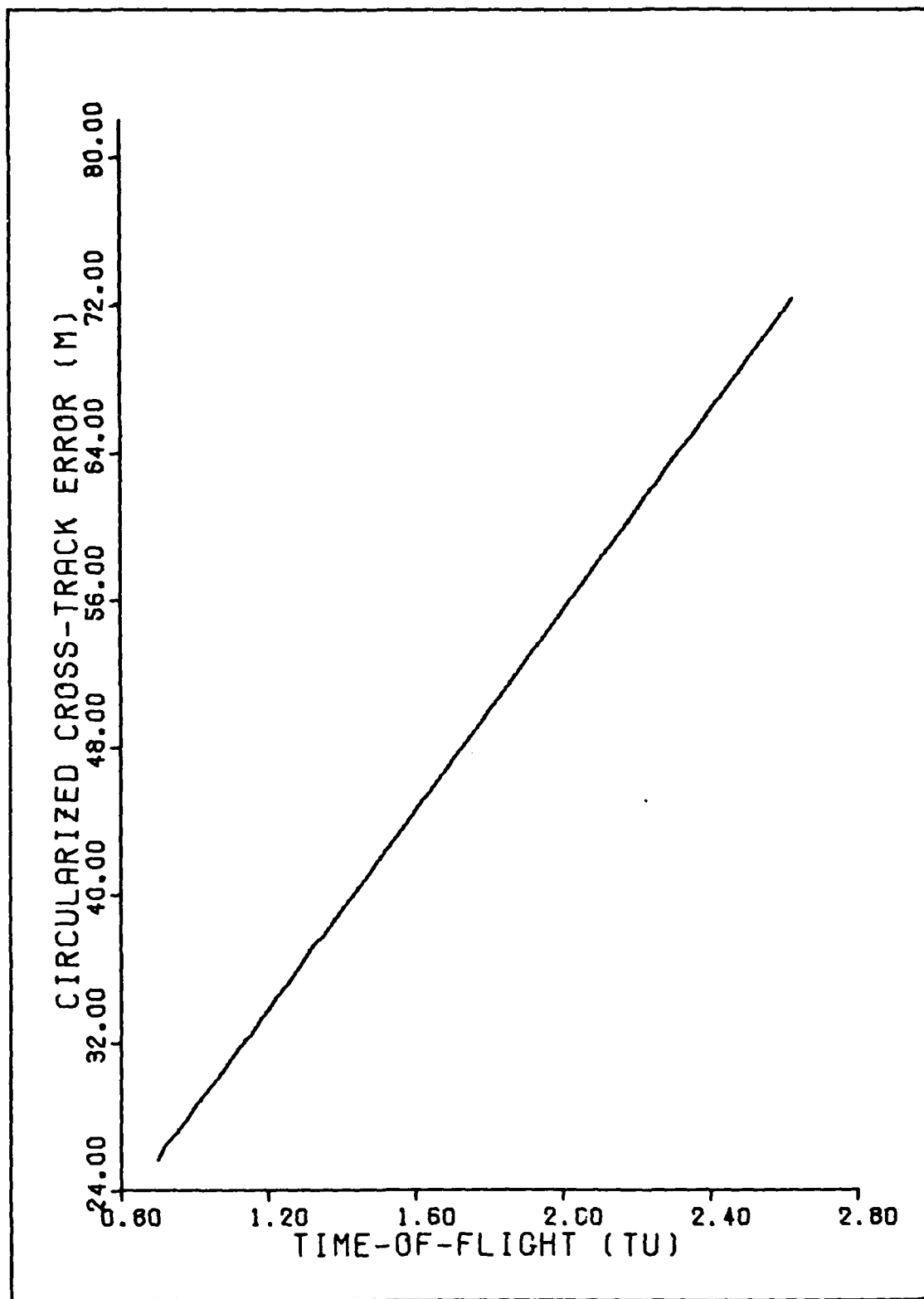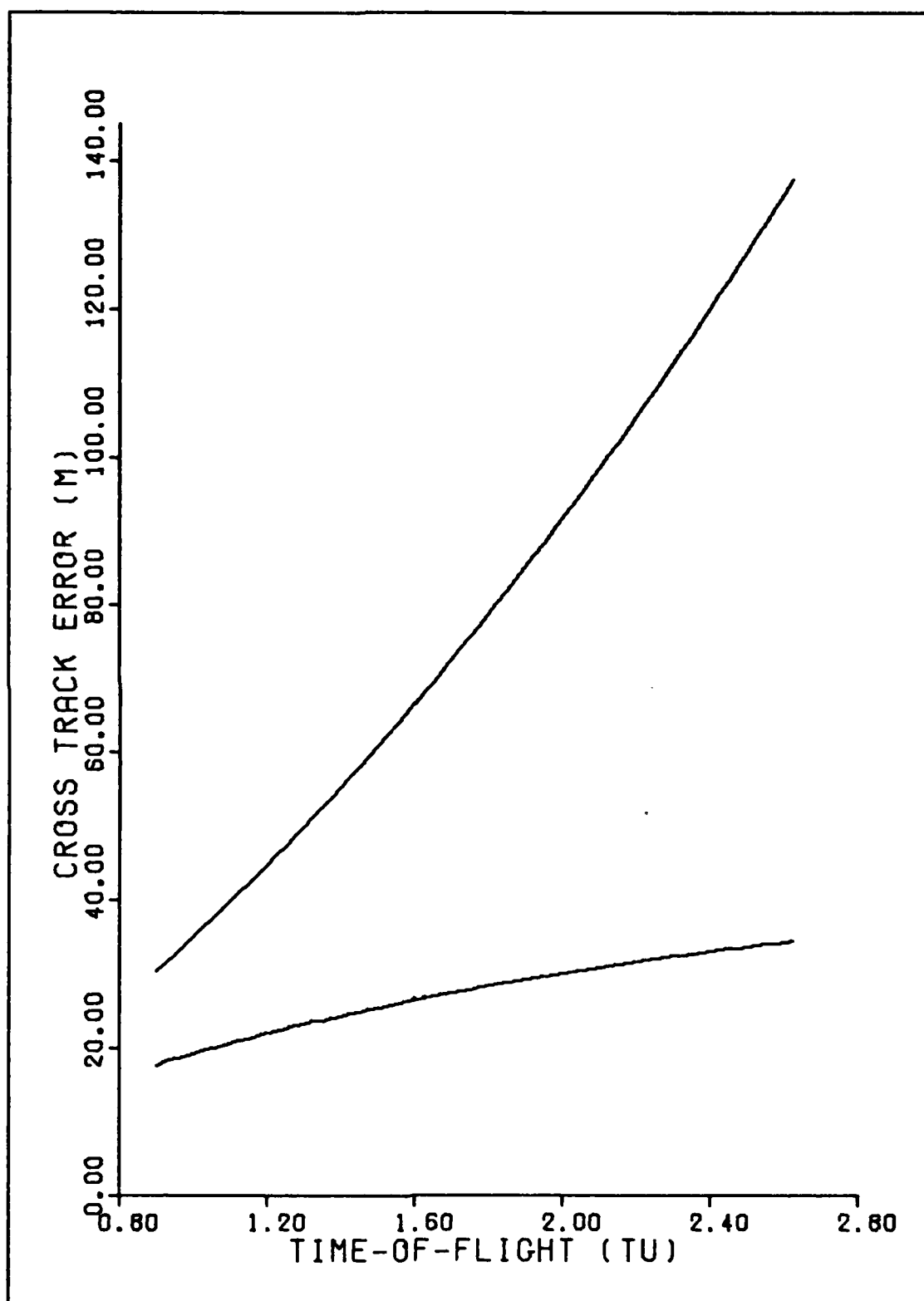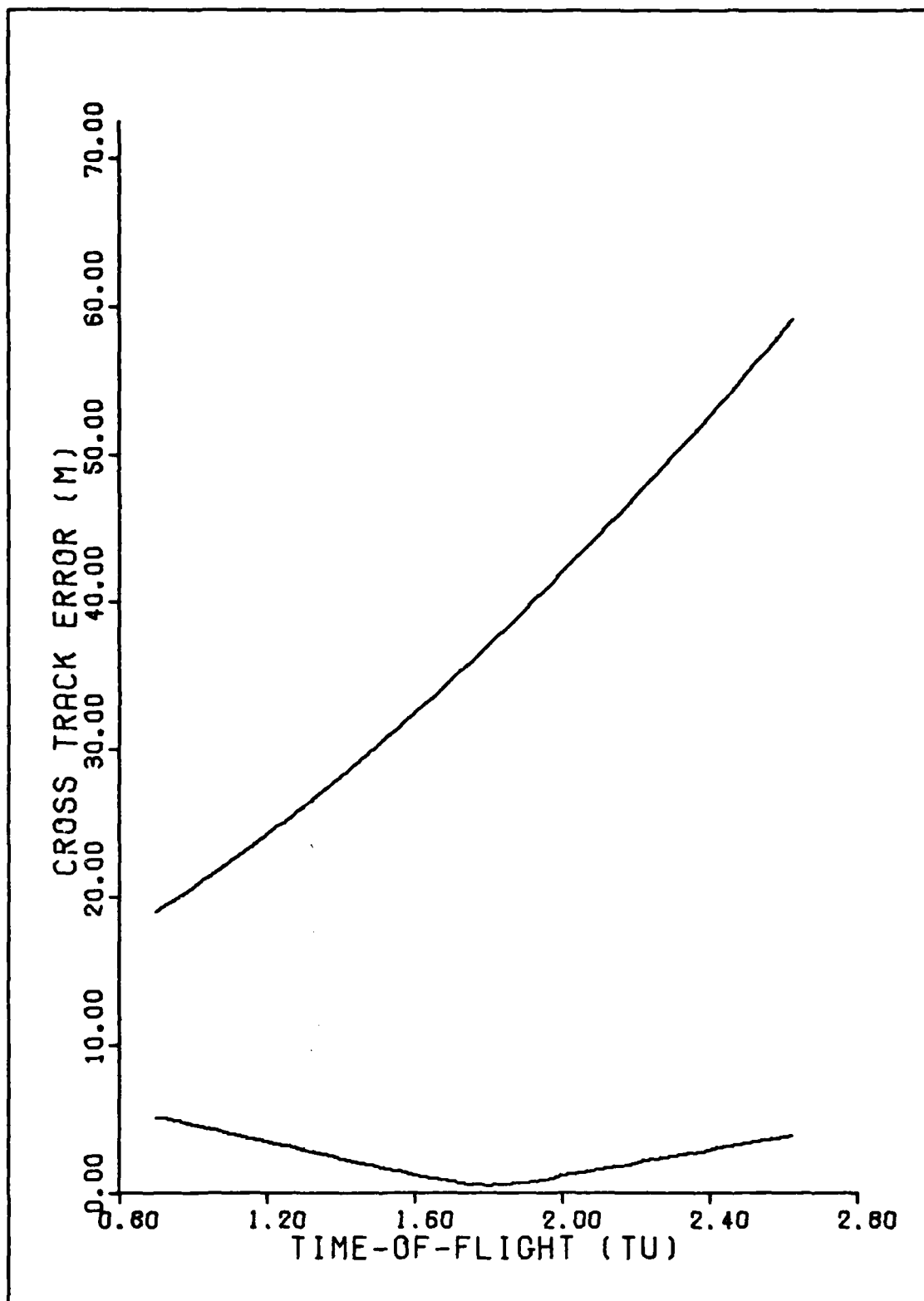
FIGURES 32 - 35.

Single Family of Intercepts

TARGET # 3.
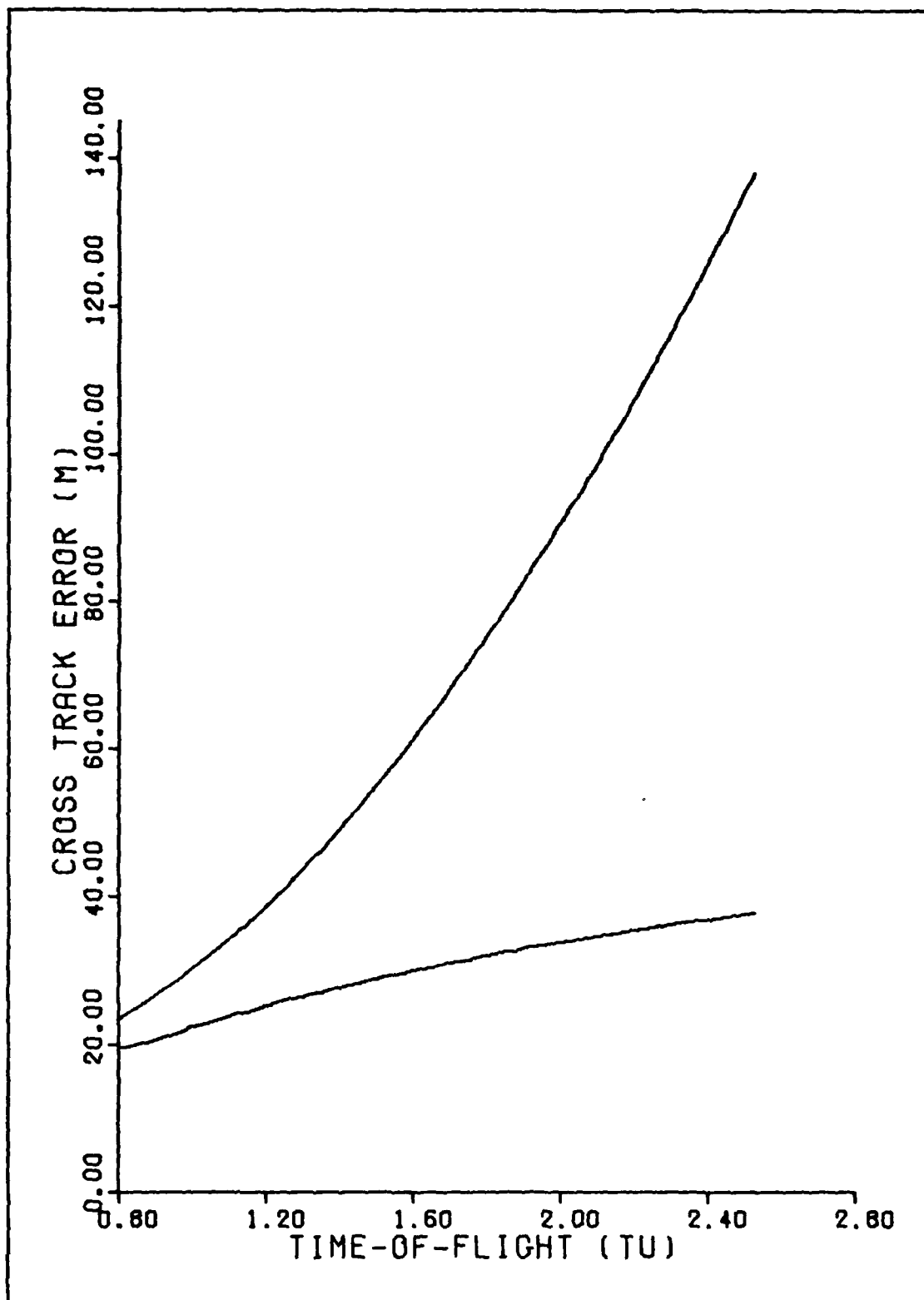
Figure # 32

Figure # 33

66

Figure # 34

67

Figure # 35

68
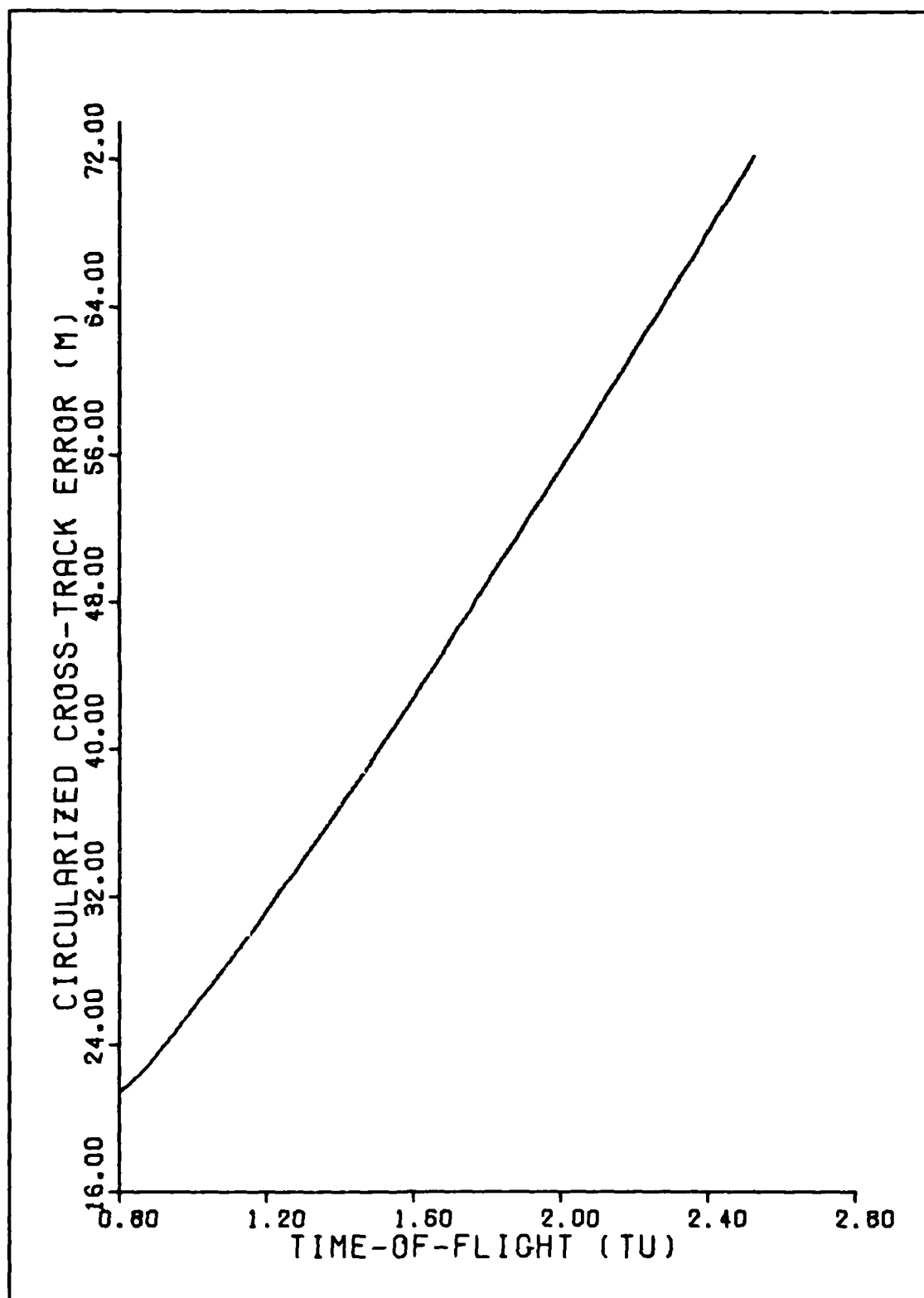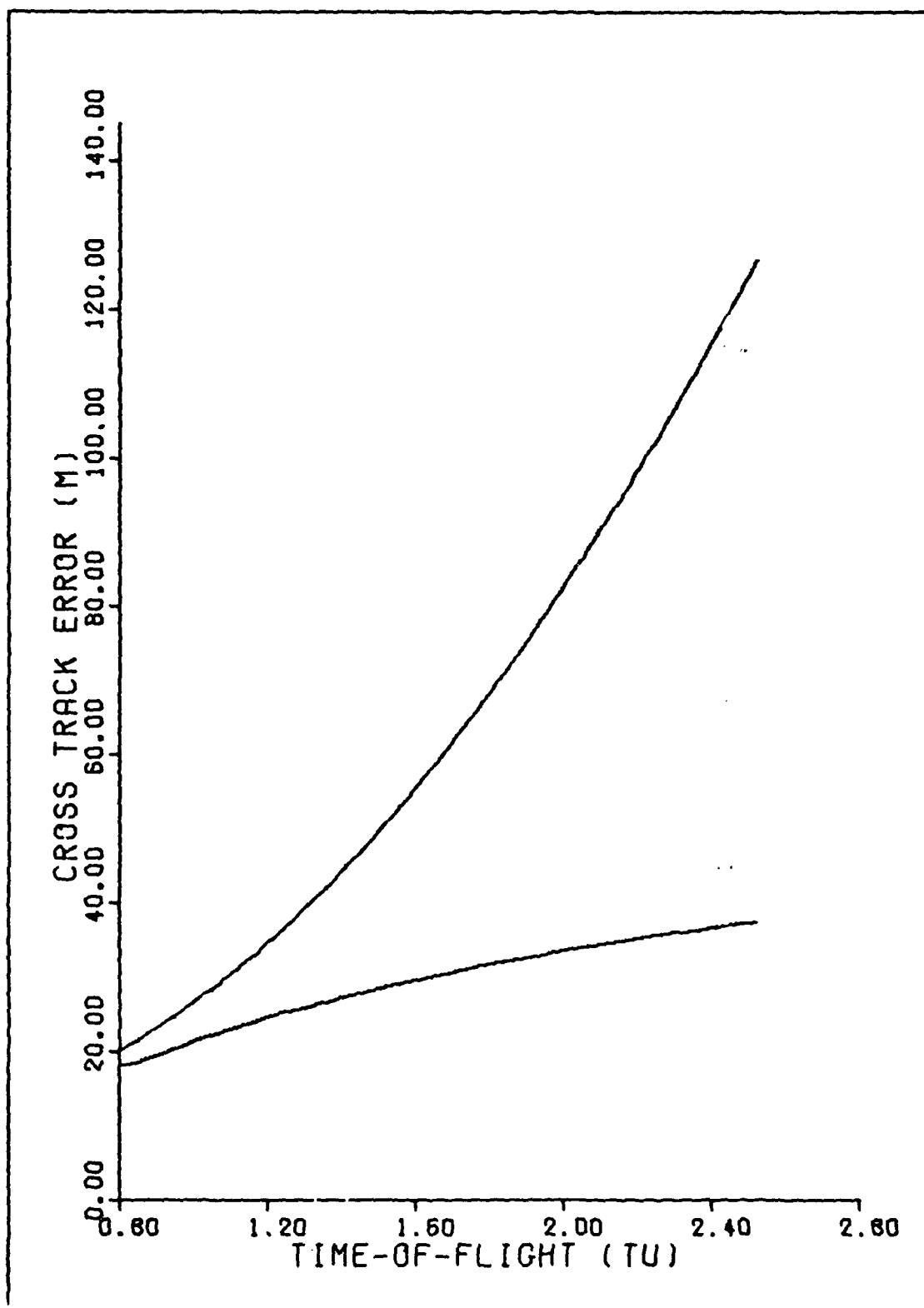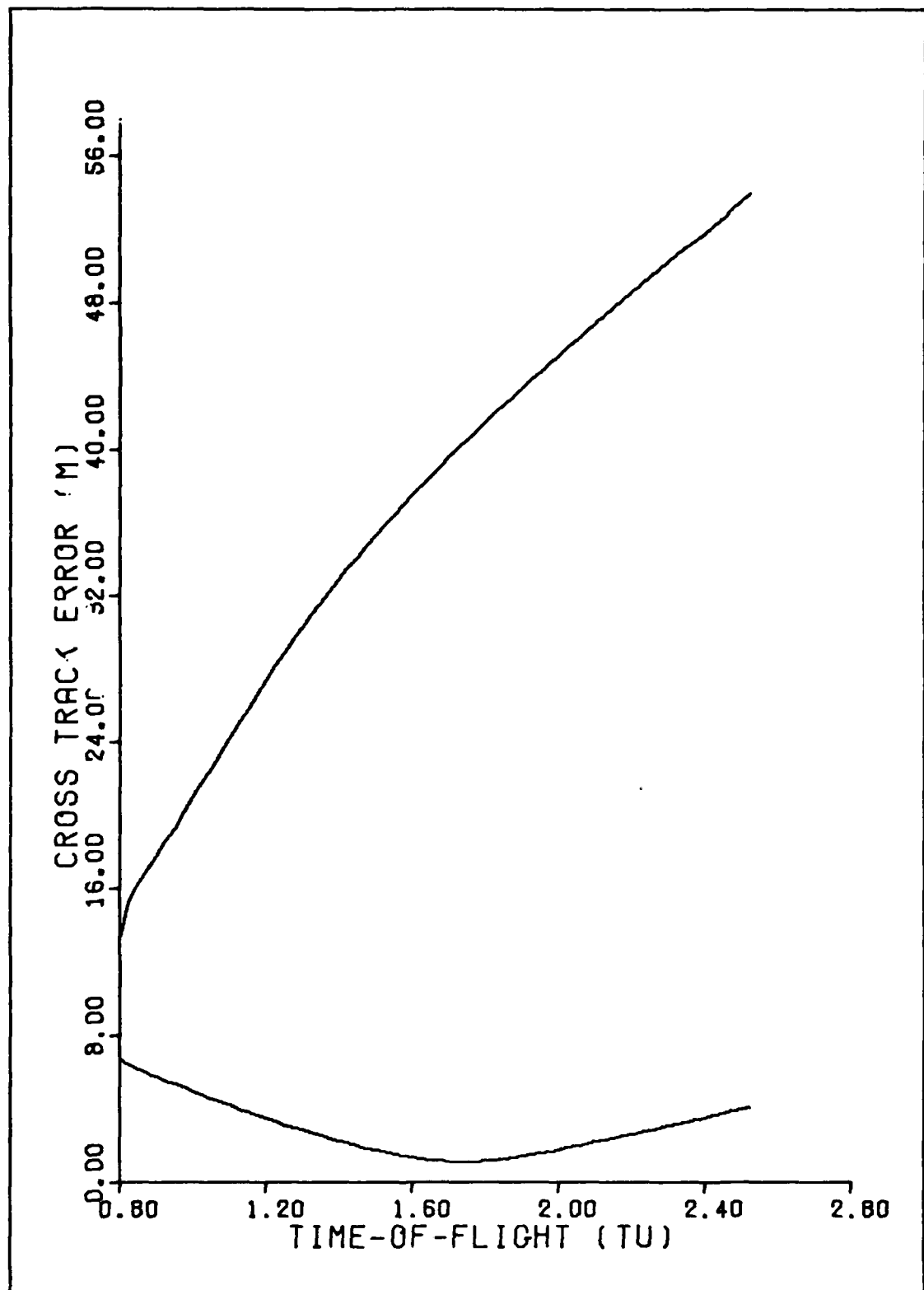
FIGURES 36 - 39.

Single Family of Intercepts

TARGET # 4.

Figure # 36

70
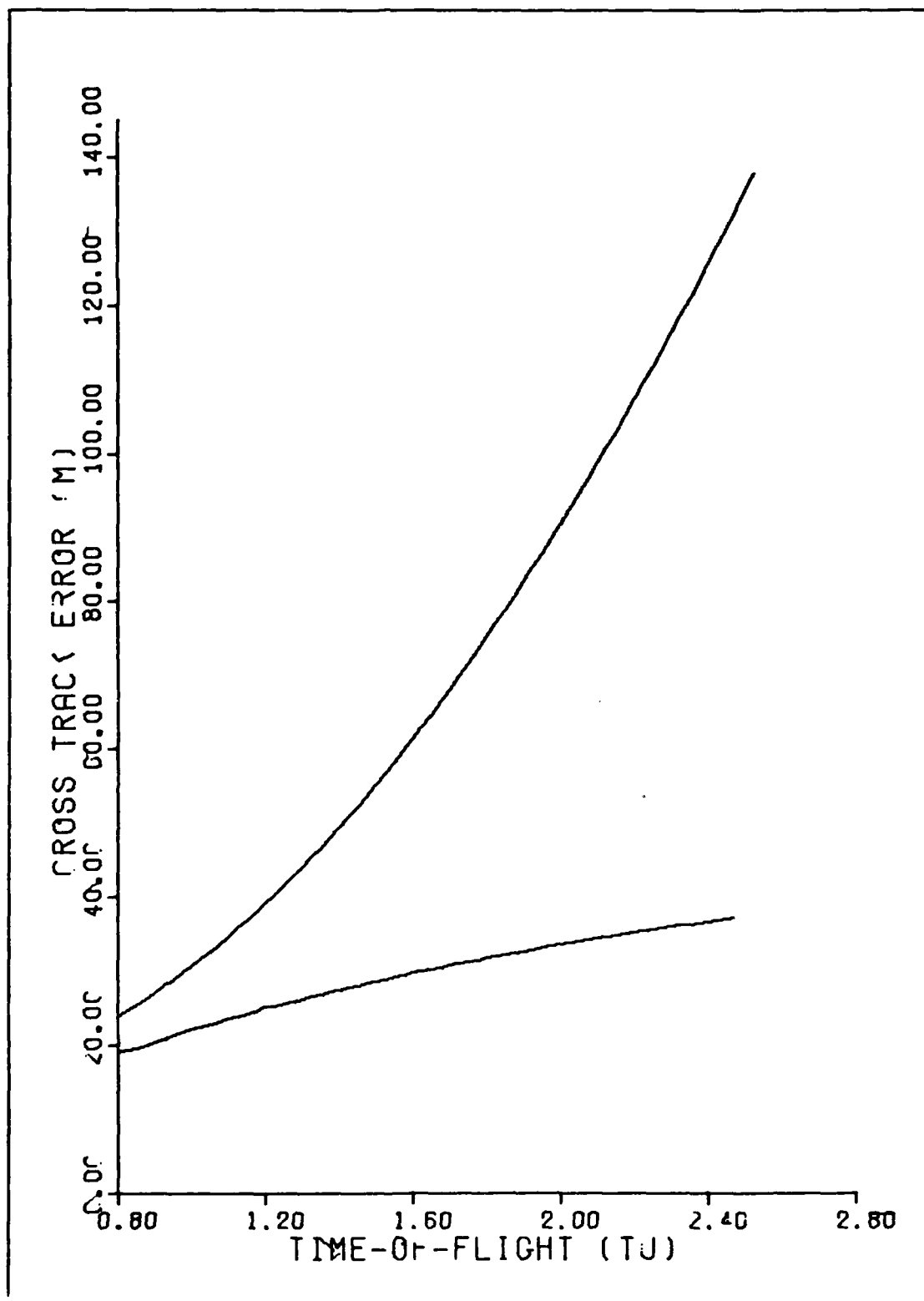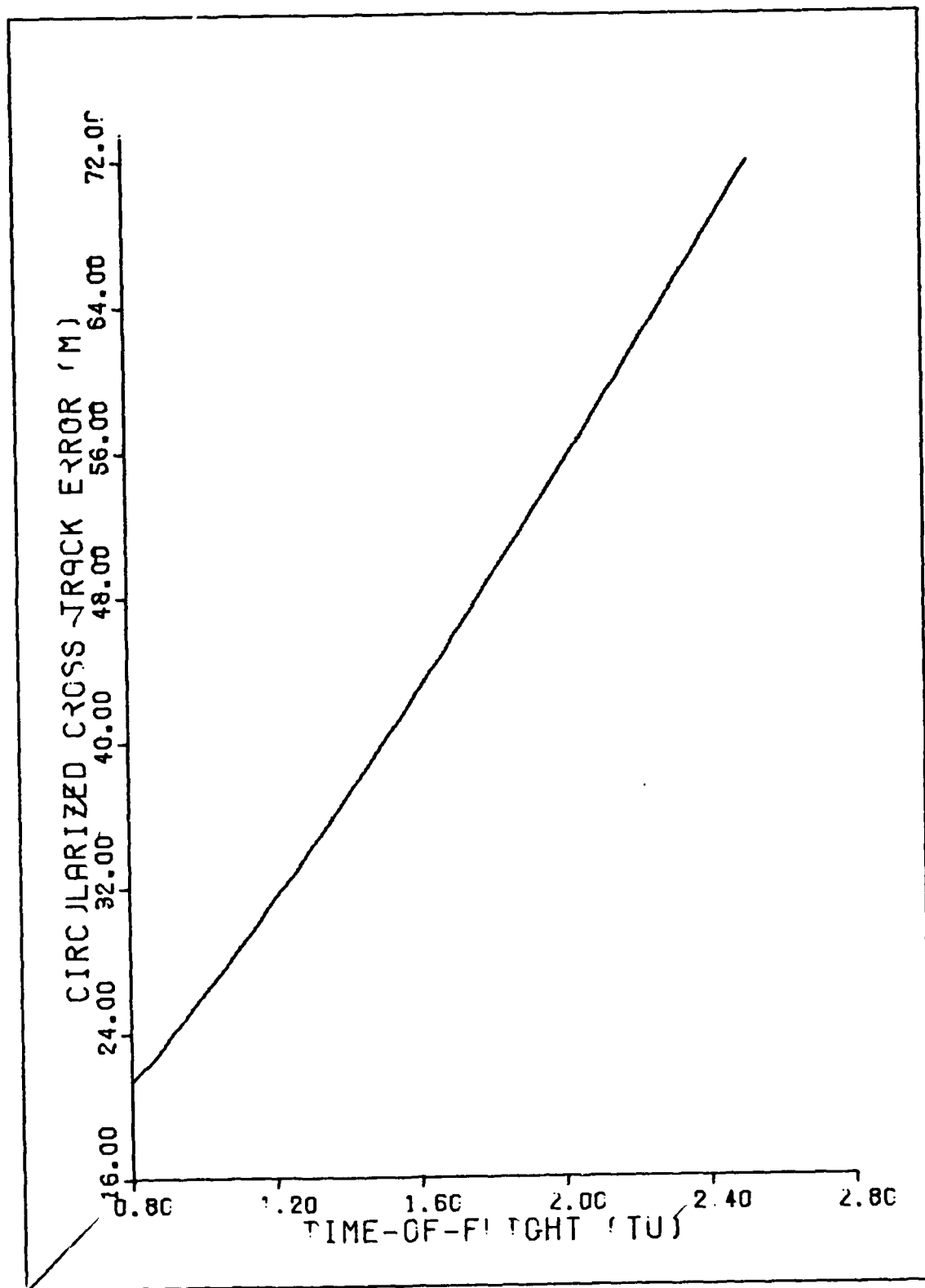
Figure # 37

Figure # 38

72

Figure # 39

73
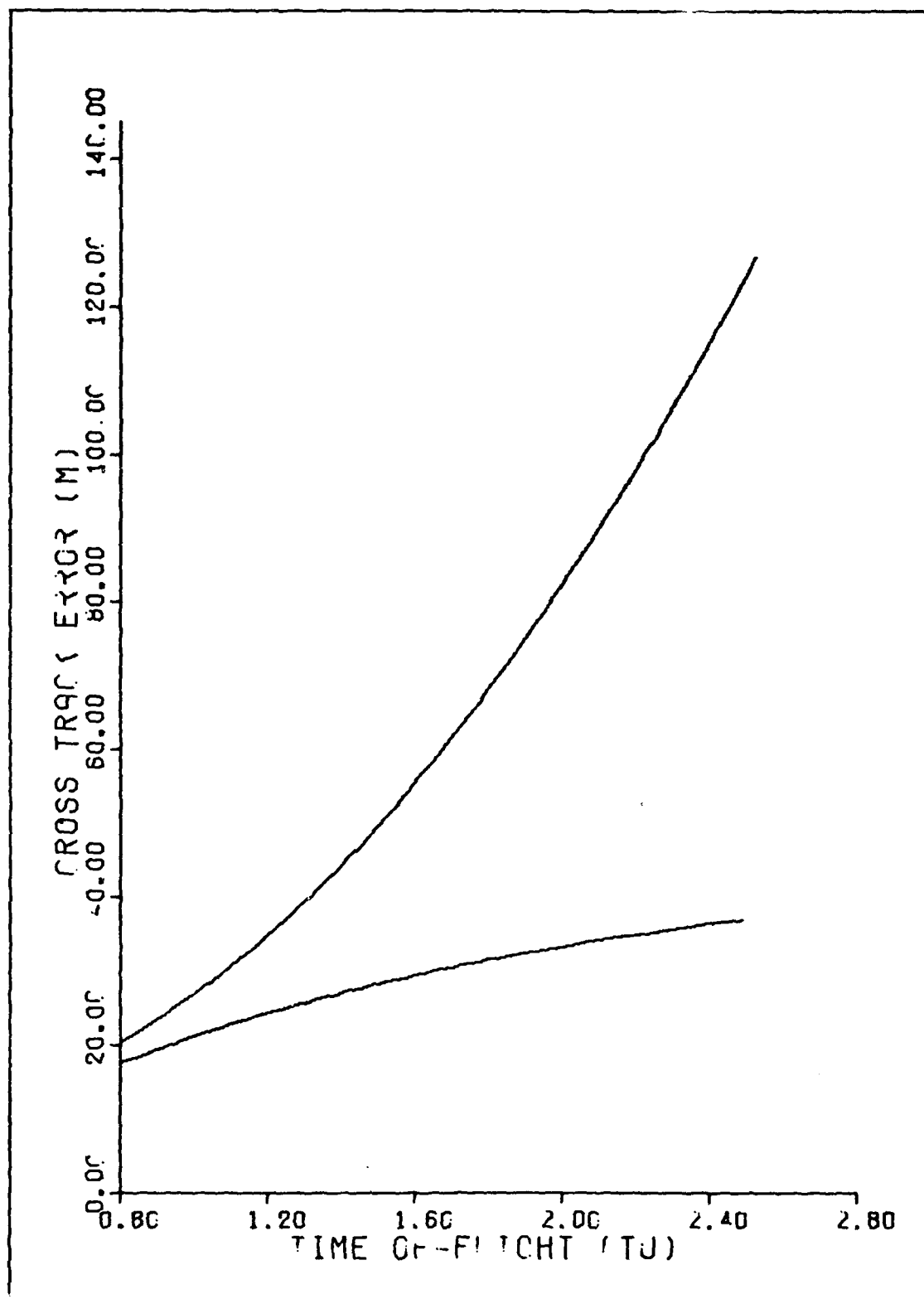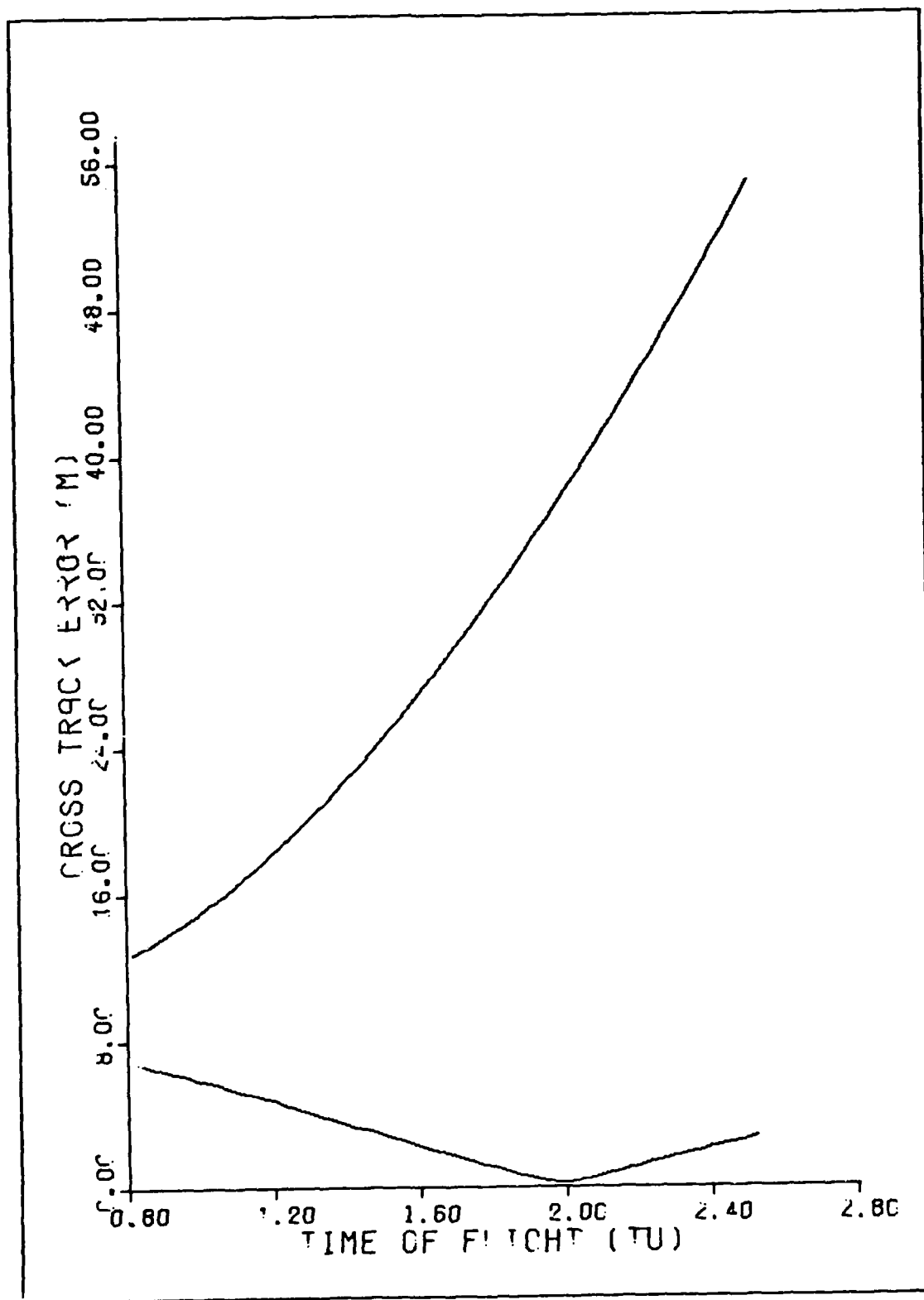
Conclusions

In analyzing the results, the first obvious conclusion is the fact that shorter times-of-flight give the smallest miss distances at the intercept point. This observation is consistant throughout the data analysis and does not depend on orbit geometry. But this fact was an expected result in that, due to the linearity of the propagation routine and a bit of common sense, the errors should increase as the interceptor tracks further down it's orbital path, as the times-of-flight become larger. In comparing like target orbits which have different eccentricities it was seen that the eccentric orbit yielded smaller intercept errors for the same time-of-flight. The length discrepencies appeared in both principal axes. This fact indicates that the resultant intercept error is dependent on the eccentricity of the target satellite's orbit. Comparing data for identical times-of-flight between the two different geometry orbits considered, it was evident that the inclined target orbit gave the smaller intercept errors. This suggests that the inclination of the target satellite has a definite effect on the final target miss distance. But a much more extensive examination along this line should be attempted before a definitive statement concerning this effect can be made.

The most generalized observation that can be made in examining the results concerns the magnitudes of the computed errors themselves. The fact that these one sigma errors, represented by the circularized error parameter, typically range between 21.3 and 74.9 meters indicates that the final free flight intercept position is known better than the position of the target itself. For the purposes of this study, the target's position at intercept was assumed to be perfectly known when in reality it's three sigma location is only known to the order of about 500 meters. Now comparing the same target orbit data while initial position and velocity errors were nulled shows that the final intercept error is mainly dependent on initial booster velocity errors and is not very sensitive to corresponding position errors at the burnout point. Finally, in examining the plot of the circularized error parameter vs. time-of-flight it is noticed that, for all the cases considered, it is a straight line increasing over time. This result was also expected, as stated previously, due to the linearity of the error propagation over time and this fact lends considerable credibility to the computed data.

Recommendations

Throughout the development of this thesis, and especially in the final stages, I became more and more aware of the many things that could, and should be done, for which I just would have time. The next step in the utilization of the program would be to alter the logic so that the covariance propagation is done from an actual burnout point to the intercept point. Because of the initial assumption of an instantaneous burn of the booster, the program propagated the covariance from the launch site to the intercept point. An appropriate burnout altitude could be inputted as a variable and it's location on the path of the intercept orbit could be found. It then remains to use this location to begin the covariance matrix propagation to give more realistic results. Another interesting area which could be examined is the development of a contour map of time of launch for the interceptor vs. cross-track error for various times-of-flight. Considering a certain launch time, corresponding launch and intercept points could be found and a family of intercept orbits computed between the points along with the associated miss distance. Then the launch time could be increased and the process repeated until a wide range of launch times have been considered. Appropriate conclusions could then be made as

to the effect of the time-of- launch on the ultimate inter-
ceptor's error at the target point. Also of interest might
be the examination of the sensitivity of the interceptor's
error to intercepts on different passes of the target rel-
ative to the launch point. Here, hopefully conclusions could
be made as to which revolution of a target should be used
to fly the most accurate intercept trajectory against.

In the implementation of this program, I see that I've
just scratched the surface by finding the cross-track errors
at the intercept point and doing the rather general orbit
geometry comparison. This groundwork has been laid in hopes
that follow-on research would be attempted to further broad-
en the knowledge on this subject.

Bibliography

1. Bate,Roger R., Mueller, Donald D.,and White, Jerry
   E. <u>Fundamentals of Astrodynamics</u>. New York. Dover
   Publications, 1971.

2. Technical Staff, The Analytic Sciences Co. <u>Applied
   Optimal Estimation</u>. Edited by Gelb, Authur. Cam-
   bridge, Massachusetts: The MIT Press

3. Wiesel, William E. Lecture Materials distributed
   in MC. 6.36, Advanced Astrodynamics. School of Eng-
   ineering, Air Force Institute of Technology, Wright-
   Patterson AFB, 1978.

4. Kaplan, Marshall H. <u>Modern Spaecraft Dynamics and
   Control</u>. New York. John Wiley and Sons, Inc. 1976.

5. Meirovitch, Leonard. <u>Methods of Analytical Dynamics</u>.
   New York. McGraw-Hill, Inc. 1970

6. Likins, Peter W. <u>Elements of Engineering Mechanics</u>.
   New York. McGraw-Hill, Inc. 1973..

7. Wiesel, William E. Lecture materials distributed in
   MC 5.32, Astrodynamics. School of Engineering, Air
   Force Institute of Technology, Wright-Patterson AFB,
   1977.

8. Reid, J. Gary Lecture materials distributed in
   EE 5.10, Linear Systems Analysis and Computational
   Methods. School of Engineering, Air Force Institute
   of Technology, Wright-Patterson AFB, 1978

9. Hornbeck, Robert W. <u>Numerical Methods</u>. New York
   Quantum Publishers, Inc. 1975

APPENDIX A

COMPUTER PROGRAM

```
      PROGRAM SAT(INPUT=/80,OUTPUT,TAPE5=OUTPUT)
      DIMENSION A(5,6),B(6,6),C(5,5),D(3,1),SS(3,1),WK(60),CHECK(6,6)
      DIMENSION PHI(6,6),RR(3,3),ROT(3,3),COV(6,6),COVR(2,2),DD(3,3)
      DIMENSION EF(3,3),U(2,2),W(2)
      DIMENSION YA(400),ZA(400),ARY(810),TT(400),DEP(400)
      COMMON THETA
      COMMON/LSTIME/LONGI,DAY,TIME,THETAG0
      COMMON/SITEE/LATI,HI
      COMMON/TIME/T
      COMPLEX U,W
      REAL LATI,LONGI,LAT,LONG
      REAL NJ,MAG1,MAG2,L1,L2
      INTEGER CD
C*********************************************************************
C*********************************************************************
C****       ERROR PROPAGATION FOR GROUND-TO-SATELLITE ELLIPTICAL     **
C**** INTERCEPT ORBITS TO EXAMINE C.E.P. GROWTH OF BOOSTER COVARIANCE **
C*********************************************************************
C*********************************************************************
      PRINT 88
      PRINT 88
88    FORMAT(1X,12)(1H*))
C
C
      KLM=0
      III=0
111   IF(III.EQ.3)GO TO 222
      KLM=KLM+1
      III=III+1
      CD=0
      LK=0
      PRINT*," "
      PRINT*,"###### ORBIT NUMBER ",III," ######"
      PRINT*," "
C*********************************************************************
C     INPUT DATA
```

A-1

```
C***********************************************************************
      READ*,T,IF,JF
      READ*,RP,VP
      READ*,LATI,LONGI,HI,TIME,DAY,THETAGO
C***********************************************************************
      PRINT*,"COVARIANCE VALUES...POSITION ERROR=",RP,"(M)"
      PRINT*,"                     VELOCITY ERROR=",VP,"(M/SEC)"
      PRINT*," "
      RP=(RP/6378145)**2
      VP=(VP/7905.36828)**2
      DO 15 I=1,3
15    COV(I,I)=RP
      DO 16 J=4,6
16    COV(J,J)=VP
      DO 9 I=1,400
      YA(I)=0.
      ZA(I)=0.
      ARY(I)=0.
9     TT(I)=0.
C***********************************************************************
C     COMPUTE LOCAL SIDERIAL TIME OF LAUNCH SITE
C***********************************************************************
      CALL LSTIME
C***********************************************************************
      T=.1
      THETA=0.
C***********************************************************************
C     COMPUTE INERTIAL COORDINATES OF LAUNCH SITE
C***********************************************************************
      CALL SITE(X,Y,Z,VX,VY,VZ)
C***********************************************************************
C     COMPUTE INITIAL SATELLITE POSITION
C***********************************************************************
      CALL ORBIT(XI,YI,ZI,VXI,VYI,VZI,RO,VO,A)
      IJ=0
      JJJ=1
```

A-2

```fortran
C**********************************************************************
C      UPDATE SATELLITE POSITION TO INTERCEPT POINT
C**********************************************************************
       DOT=XI*VXI+YI*VYI+ZI*VZI
76     CALL FINDX(XX,DOT,RO,VO,IND,A)
       IF(IND.EQ.50)GO TO 19
       CALL LOCATE(XX,RO,VO,F,G,S,CC,ZZ,A)
       R1=F*XI+G*VXI
       R2=F*YI+G*VYI
       R3=F*ZI+G*VZI
       DF=XX*(ZZ*S-1.)/(R*RO)
       DG=1.-XX**2.*CC/R
       V4=DF*XI+DG*VXI
       V5=DF*YI+DG*VYI
       V6=DF*ZI+DG*VZI
       R=SQRT(R1**2.+R2**2.+R3**2.)
       V=SQRT(V5**2+V6**2+V4**2)
C**********************************************************************
C      COMPUTE ORBIT REQUIRED FOR PROBE INTERCEPT
C**********************************************************************
       MAG1=SQRT(X**2.+Y**2.+Z**2.)
       MAG2=SQRT(R1**2.+R2**2.+R3**2.)
       X1=X/MAG1
       Y1=Y/MAG1
       Z1=Z/MAG1
       X2=R1/MAG2
       Y2=R2/MAG2
       Z2=R3/MAG2
       DOP=X1*X2+Y1*Y2+Z1*Z2
       NU=ACOS(DOP)
       CALL GAUSS(NU,MAG1,MAG2,F,G,DG,LM)
       IF(LM.EQ.1)GO TO 77
       IF(LM.EQ.0)GO TO 77
       VI1=(R1-F*X)/G
       VI2=(R2-F*Y)/G
       VI3=(R3-F*Z)/G
```

A-3

```
          VT1=(DG*R1-X)/G
          VT2=(DG*R2-Y)/G
          VT3=(DG*R3-Z)/G
          VI=SQRT(VI1**2.+VI2**2.+VI3**2.)
          VVS=SQRT(VT1**2+VT2**2+VT3**2)
          VVI=SQRT(VI1**2.+VI2**2.+VI3**2.)
          DOT1=X*VI1+Y*VI2+Z*VI3
          CC=VI**2.-1./MAG1
          E1=CC*X-DOT1*VI1
          E2=CC*Y-DOT1*VI2
          E3=CC*Z-DOT1*VI3
          EU=SQRT(E1**2.+E2**2.+E3**2.)
          TI=T
          AN=ABS(1./(VI**2-2./MAG1))
C*******************************************************************
C      BRANCH--ELLIPTICAL OR HYPERBOLIC INTERCEPT ORBIT
C*******************************************************************
          IF(EU.GT.1)GO TO 50
          GO TO 500
229       PRINT*,"NO ELLIPTICAL INTERCEPT ORBITS POSSIBLE FOR GIVEN DATA"
          STOP
50        IF(KK.EQ.1)GO TO 111
77        T=T+.1
          IF(T.GT.20.)GO TO 229
          GO TO 78
500       CONTINUE
C*******************************************************************
C      DEVELOP THE STATE TRANSITION MATRIX
C*******************************************************************
51        JM=1
          LL=0
          I=1
          TW=.0001
          XW=X+TW
          YW=Y+TW
          ZW=Z+TW
```

A-4

```
          V1W=VI1+TW
          V2W=VI2+TW
          V3W=VI3+TW
          RI1=R1
          RI2=R2
          RI3=R3
          VIS1=VT1
          VIS2=VT2
          VIS3=VT3
          XC=XW
          YC=Y
          ZC=7
          V1C=VI1
          V2C=VI2
          V3C=VI3
          RO=SQRT(XW**2+Y**2+7**2)
          VO=SQRT(VI1**2+VI2**2+VI3**2)
          DOT=XW*VI1+Y*VI2+Z*VI3
99        A=1./(VO**2-2./RO)
          A=ABS(A)
          CALL FINDY(XX,DOT,RO,VO,IVD,A)
          CALL LOCATE(XX,RO,VO,F,G,S,CC,ZZ,A)
          R1=F*XC+G*V1C
          R2=F*YC+G*V2C
          R3=F*ZC+G*V3C
          R=SQRT(R1**2.+R2**2.+R3**2.)
          DF=XX*(ZZ*S-1.)/(R*RO)
          DG=1.-XX**2.*CC/R
          V1=DF*XC+DG*V1C
          V2=DF*YC+DG*V2C
          V3=DF*ZC+DG*V3C
          V=SQRT(V1**2.+V2**2.+V3**2.)
          CHECK(1,I)=(R1-RI1)/TW
          CHECK(2,I)=(R2-RI2)/TW
          CHECK(3,I)=(R3-RI3)/TW
          CHECK(4,I)=(V1-VIS1)/TW
```

```
            CHECK(5,I)=(V2-VIS2)/TW
            CHECK(5,I)=(V3-VIS3)/TW
            IF(LL.EQ.1)GO TO 96
            IF(I.EQ.1)GO TO 72
            IF(I.EQ.2)GO TO 73
            IF(I.EQ.3)GO TO 74
            IF(I.EQ.4)GO TO 75
            IF(I.EQ.5)GO TO 76
            IF(I.EQ.6)GO TO 96
      72    I=2
            RO=SQRT(X**2+YW**2+Z**2)
            DOT=X*VI1+YW*VI2+Z*VI3
            XC=X
            YC=YW
            GO TO 99
      73    I=3
            RO=SQRT(ZW**2+Y**2+X**2)
            DOT=X*VI1+Y*VI2+ZW*VI3
            YC=Y
            ZC=ZW
            GO TO 99
      74    I=4
            RO=SQRT(X**2+Y**2+Z**2)
            VO=SQRT(V1W**2+VI2**2+VI3**2)
            DOT=X*V1W+Y*VI2+Z*VI3
            ZC=Z
            V1C=V1W
            GO TO 99
      75    I=5
            VC=SQRT(VI1**2+V2W**2+VI3**2)
            DOT=X*VI1+Y*V2W+Z*VI3
            V1C=VI1
            V2C=V2W
            GO TO 99
      76    I=6
            VC=SQRT(VI1**2+VI2**2+V3W**2)
```

1-6

```
        DOT=X*VI1+Y*VI2+Z*V3W
        V2C=VI2
        V3C=V3W
        LL=1
        GO TO 99
96      CONTINUE
C+*******************************************************************
C       COMPUTE UPDATED COVARIANCE MATRIX
C+*******************************************************************
        CALL MULT(CHECK,COV,C,6,6,6)
        DO 7 I=1,6
        DO 7 J=1,6
7       B(I,J)=CHECK(J,I)
        CALL MULT(C,B,PHI,6,6,6)
C+*******************************************************************
C       EXTRACT POSITION SUB-ELEMENTS OF COVARIANCE MATRIX
C+*******************************************************************
        DO 2 I=1,3
        DO 1 J=1,3
1       RK(I,J)=PHI(I,J)
2       CONTINUE
C+*******************************************************************
C       ALIGN COVARIANCE ELEMENTS ALONG SATELLITE TRACK
C+*******************************************************************
        VXR=VT1-V4
        VYR=VT2-V5
        VZR=VT3-V6
        VR=SQRT(VXR**2+VYR**2+VZR**2)
        VXY=SQRT(VXR**2+VYR**2)
        CPX=VXR/VXY
        SPX=VYR/VXY
        PZ=ATAN(VZR/VXY)
        ROT(1,1)=COS(PZ)*CPX
        ROT(1,2)=COS(PZ)*SPX
        ROT(1,3)=-SIN(PZ)
        ROT(2,1)=-SPX
```

A-7

```fortran
      ROT(2,2)=CPX
      ROT(2,3)=0.
      ROT(3,1)=SIN(PZ)*CPX
      ROT(3,2)=SIN(PZ)*SPX
      ROT(3,3)=COS(PZ)
      CALL MULR(ROT,RR,DD,3,3,3)
      DO 8 I=1,3
      DO 9 J=1,3
8     EE(I,J)=ROT(J,I)
      CALL MULR(DD,EE,EE,3,3,3)
      DO 11 I=1,2
      DO 10 J=1,2
10    COVR(I,J)=EE(I+1,J+1)
11    CONTINUE
C*******************************************************************************
C     COMPUTE ERROR ELLIPSOID RESULTS
C*******************************************************************************
      I7=2
      CALL EIGRF(COVR,2,2,2,W,U,IZ,WK,IER)
      DOT=V4*VT1/(V*VVS)+V5*VT2/(V*VVS)+V6*VT3/(V*VVS)
      HANG=ACOS(DOT)
      HANG=HANG*180./3.14159265359
      L1=W(1)
      L2=W(2)
      IF(L1*L2.LT.0.)GO TO 111
      L1=6378145.*SQRT(L1)
      L2=6378145.*SQRT(L2)
      TM=T*13.44683457
66    CONTINUE
      CEP=SQRT(L1*L2)
      QVI=VI*7.90536828
      LK=LK+1
      PRINT*,"--------------------------------------------"
      PRINT*,"TIME TO INTERCEPT=",TM,"(MIN)    ",T,"(TU)"
      PRINT*,"ELLIPSOID AXES LENGTHS ",L1,"(M)"
      PRINT*,"                            ",L2,"(M)"
```

A-8

```
        PRINT*,"INTERCEPT ANGLE=",HANG,"(DEG)"
        PRINT*,"INTERCEPTOR LAUNCH VELOCITY=",OVI,"(KM/SEC)"
        PRINT*,"***CEP=",CEP,"(M)"
        IF(LK.GT.70)GO TO 27
        CD=CD+1
        I=2*CD-1
        J=2*CD
        YA(CD)=L1
        ZA(CD)=L2
        TT(CD)=T
        ARY(I)=L1
        ARY(J)=L2
        DEP(CD)=CEP
100     T=T+IF*.025
        KK=1
        JJJ=JJJ+1
        GO TO 78
27      CALL PLOTR(YA,ZA,TT,ARY,CD)
        IF(III.EQ.1)GO TO 66
        GO TO 111
66      CALL ERR(DEP,TT,CD)
        GO TO 111
222     IF(KLM.EQ.12)GO TO 223
        III=0
        PRINT*," "
        PRINT*," "
        PRINT*," "
        GO TO 111
223     CALL PLOTE
        STOP
        END
```

A-9

```
      SUBROUTINE LSTIME
      COMMON THETA
      COMMON/LSTIME/LONGI,DAY,TIME,THETAGO
      REAL LONGI,LONG
      PI=3.141592654
      LONG=LONGI*PI/180.
      TIME=TIME/2400.
      DAY=DAY+TIME
1     THETA=THETAGO+1.0027379003*2.*PI*DAY+LONG
      RETURN
      END
```

```
SUBROUTINE SITE(X,Y,Z,VX,VY,VZ)
COMMON THETA
COMMON/SITEE/LATI,HI
REAL LAT,LATI
AE=1.000003136
BE=.9956710539
E=.08191
PI=3.141592654
LAT=LATI*PI/180.
H=HI/2.092567257E7
XI=(AE/(1.-E**2.*SIN(LAT)**2.)**.5+H)*COS(LAT)
Z=(AE*(1.-E**2.)/(1.-E**2.*SIN(LAT)**2.)**.5+H)*SIN(LAT)
X=XI*COS(THETA)
Y=XI*SIN(THETA)
VEARTH=.0588336001
VX=-VEARTH*Y
VY=VEARTH*X
VZ=0.
RETURN
END
```

```
      SUBROUTINE MILT(Q,B,C,L,M,N)
      DIMENSION Q(5,6),B(6,6),C(5,5),AD(5,6),BD(6,6),CD(6,6)
100   DO 108 J=1,M
      DO 104 I=1,L
104   AD(I,J)=Q(I,J)
      DO 108 K=1,N
108   BD(J,K)=B(J,K)
      DO 112 I=1,L
      DO 112 J=1,N
      CD(I,J)=0.
      DO 112 K=1,M
112   CD(I,J)=CD(I,J)+AD(I,K)*BD(K,J)
      DO 116 I=1,L
      DO 116 J=1,N
116   C(I,J)=CD(I,J)
      RETURN
      END
```

```
      SUBROUTINE MULP(A,P,O,L,M,N)
      DIMENSION A(3,3),P(3,3),O(3,3),AR(3,3),BC(3,3),CB(3,3)
200   DO 208 J=1,M
      DO 204 I=1,L
204   AR(I,J)=A(I,J)
      DO 208 K=1,N
208   BC(J,K)=P(J,K)
      DO 212 I=1,L
      DO 212 J=1,N
      CB(I,J)=0.
      DO 212 K=1,M
212   CB(I,J)=CB(I,J)+AR(I,K)*BC(K,J)
      DO 216 I=1,L
      DO 216 J=1,N
216   O(I,J)=CB(I,J)
      RETURN
      END
```

```
      SUBROUTINE ORBIT(XI,YI,ZI,VXI,VYI,VZI,RO,VO,A)
      DIMENSION Q(6,6),B(6,6),C(5,5),D(3,1),SS(3,1)
      REAL I
      READ*,A,E,I,OMEGA,WMEGA,ANOMY
      IF(EOF(5LINPUT).NE.0)STOP
      PRINT*,"SATELLITE'S ORBITAL ELEMENTS (A,E,I,OMEGA,WMEGA,ANOMALY)"
      PRINT*,"##########  ",A,E,I,OMEGA,WMEGA,ANOMY," ##########"
      P=A*(1.-E**2)
      Z=3.141592653359/180.
      I=I*Z
      OMEGA=OMEGA*Z
      WMEGA=WMEGA*Z
      ANOMY=ANOMY*Z
      R=P/(1.+E*COS(ANOMY))
      B(1,1)=R*COS(ANOMY)
      B(2,1)=R*SIN(ANOMY)
      B(3,1)=0.
      D(1,1)=-(1./P**.5*SIN(ANOMY))
      D(2,1)=1./P**.5*(E+COS(ANOMY))
      D(3,1)=0.
      Q(1,1)=COS(OMEGA)*COS(WMEGA)-SIN(OMEGA)*SIN(WMEGA)*COS(I)
      Q(1,2)=-COS(OMEGA)*SIN(WMEGA)-SIN(OMEGA)*COS(WMEGA)*COS(I)
      Q(1,3)=SIN(OMEGA)*SIN(I)
      Q(2,1)=SIN(OMEGA)*COS(WMEGA)+COS(OMEGA)*SIN(WMEGA)*COS(I)
      Q(2,2)=-SIN(OMEGA)*SIN(WMEGA)+COS(OMEGA)*COS(WMEGA)*COS(I)
      Q(2,3)=-COS(OMEGA)*SIN(I)
      Q(3,1)=SIN(WMEGA)*SIN(I)
      Q(3,2)=COS(WMEGA)*SIN(I)
      Q(3,3)=COS(I)
      CALL MULT(Q,B,C,3,3,1)
      DO 11 K=1,3
      SS(K,1)=C(K,1)
11    B(K,1)=D(K,1)
      CALL MULT(Q,B,C,3,3,1)
      XT=SS(1,1)
      Y1=SS(2,1)
```

A-14

```
ZI=SS(3,1)
VXI=C(1,1)
VYI=C(2,1)
VZI=C(3,1)
RO=(XI**2.+YI**2.+ZI**2.)**.5
VO=(VXI**2.+VYI**2.+VZI**2.)**.5
A=1./(VO**2-2./RO)
A=ABS(A)
RETURN
END
```

```
      SUBROUTINE FINDX(XX,DOT,RO,VO,IND,A)
      COMMON/TIME/T
      X=T/A
      I=0
1     Z=X**2./A
      C=(1.-COS(SQRT(Z)))/Z
      S=(SQRT(Z)-SIN(SQRT(Z)))/SQRT(Z**3.)
      TT=DOT*X**2.*C+(1.-RO/A)+X**3.*S+RO*X
      DT=X**2*C+DOT*X*(1.-Z*S)+RO*(1.-Z*C)
      DELT=T-TT
      I=I+1
      IF(I.EQ.50)GO TO 10
      IF(ABS(DELT).LT.1.E-6)GO TO 20
      X=X+DELT/DT
      GO TO 1
10    PRINT*,"CONVERGENCE NOT MADE--X=",X
20    XX=X
      IND=I
      RETURN
      END
```

```fortran
SUBROUTINE LOCATE(XX,RO,VJ,F,G,S,CC,ZZ,A)
COMMON/TIME/T
Z=XX**2./A
S=(SQRT(Z)-SIN(SQRT(Z)))/SQRT(Z**3.)
C=(1.-COS(SQRT(Z)))/Z
F=1.-XX**2*C/RO
G=T-XX**3.*S
CC=C
ZZ=Z
RETURN
END
```

```
      SUBROUTINE GAUSS(NU,MAG1,MAG2,F,G,JS,LM)
      COMMON/TIME/T
      REAL NU,MAG1,MAG2
      AA=SQRT(MAG1*MAG2)*SIN(NU)/SQRT(1.-COS(NU))
      LM=0
      Z=0.
      I=0
1     IF(Z.GT..25)GO TO 55
      IF(Z.LT.-.25)GO TO 57
      GO TO 59
55    C=(1.-COS(SQRT(Z)))/Z
      S=(SQRT(Z)-SIN(SQRT(Z)))/SQRT(Z**3)
      DC=(1.-Z*S-2.*C)/(2.*Z)
      DS=(C-3.*S)/(2.*Z)
      GO TO 50
57    C=(1.-COSH(-Z))/Z
      S=(SINH(SQRT(-Z))-SQRT(-Z))/SQRT((-Z)**3)
      DC=1./24.+2.*Z/720.-3.*Z**2/40320.+4.*Z**3/3628800.
      DS=1./120.+2.*Z/5040-3.*Z**2/362880.+4. Z**3/39916800.
      GO TO 50
59    C=1./2.-Z/24.+Z**2/720.-Z**3/40320.
      S=1./6.-Z/120.+Z**2/5040.-Z**3/362880.
      DC=1./24.+2.*Z/720.-3.*Z**2/40320.+4.*Z**3/3628800.
      DS=1./120.+2.*Z/5040.-3.*Z**2/362880.+4.*Z**3/39916800.
      GO TO 50
50    I=I+1
4     Y=MAG1+MAG2-AA*(1.-Z*S)/SQRT(C)
      IF(Y.LT.0)GO TO 68
      X=SQRT(Y/C)
      TT=X**3*S+AA*SQRT(Y)
      DELT=T-TT
      DELS=ABS(DELT)
      IF(DELS.LT.1.E-6)GO TO 10
      DT=X**3*(DS-3.*S*DC/(2.*C))+AA/8.*(3.*S*SQRT(Y)/C+AA/X)
      Z=Z+DELT/DT
      IF(I.GT.500)GO TO 80
```

A-18

```
         GO TO 1
   10    F=1.-Y/MAG1
         IF(7.LE.0.)GO TO 88
         G=AA*SORT(Y)
         DG=1.-Y/MAG2.
         LM=2
         GO TO 33
   80    LM=0
         GO TO 33
   88    LM=1
   83    RETURN
         END
```

```
SUBROUTINE PLOTF(YA,ZA,TT,ARY,CD)
INTEGER CD,CP
DIMENSION YA(400),ZA(400),TT(400),ARY(810)
CD1=CD+1
CD2=CD+2
CP=2*CD
CALL PLOT(8.,-2.5,-3)
CALL PLOT(0.,1.775,-3)
CALL PLOT(0.,8.75,-2)
CALL PLOT(6.25,0.,-2)
CALL PLOT(0.,-8.75,-2)
CALL PLOT(-6.25,0.,-2)
CALL PLOT(.75,.75,-3)
CALL SCALE(TT,5.,CD,1)
CALL SCALE(ARY,7.25,CP,1)
YA(CD+1)=ARY(CP+1)
YA(CD+2)=ARY(CP+2)
ZA(CD+1)=ARY(CP+1)
ZA(CD+2)=ARY(CP+2)
CALL AXIS(0.,0.,19HTIME-OF-FLIGHT (TU),-19,5.,0.,TT(CD1),TT(CD2))
CALL AXIS(0.,0.,21HCROSS TRACK ERROR (M),21,7.25,90.,ARY(CP+1),ARY
*(CP+2))
CALL LINE(TT,YA,CD,1,0,0)
CALL LINE(TT,ZA,CD,1,0,0)
RETURN
END
```

```
      SUBROUTINE ERR(DEP,TT,CO)
      DIMENSION DEP(400),TT(400)
      INTEGER CO
      CALL PLOT(8.,-2.5,-3)
      CALL PLOT(0.,1.375,-3)
      CALL PLOT(0.,8.75,-2)
      CALL PLOT(5.25,0.,-2)
      CALL PLOT(0.,-8.75,-2)
      CALL PLOT(-5.25,0.,-2)
      CALL PLOT(.75,.75,-3)
      CALL SCALE(TT,5.,CO,1)
      CALL SCALE(DEP,7.25,CO,1)
      CALL AXIS(0.,0.,19HTIME-OF-FLIGHT (TU),-19,5.,0.,TT(CO+1),TT(CO+2
     *)
      CALL AXIS(0.,0.,34HCIRCULARIZED CROSS-TRACK ERROR (M),34,7.25,90.
     $DEP(CO+1),DEP(CO+2))
      CALL LINE(TT,DEP,CO,1,0,0)
      RETURN
      END
```

A-21

Vita

Michael J. VonPlinsky was born on 24 September 1955, in Rochester, New York. After graduation from High School, he attended the Pennsylvania State University, receiving a Bachelor of Science degree in Aerospace Engineering on 28 May 1977. Upon graduation, he was commissioned in the United States Air Force. In September 1977, he entered the Air Force Institute of Technology, Wright Patterson AFB, Ohio, as a resident graduate student in Astronautics.

Permanent Address: 1339 Towerlawn Drive
                    Monroeville, PA 15146

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GA/AA/78D-8	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Error Propagation for Ground-to- Satellite Elliptical Intercept Orbits to Examine the Error Ellipsoid Growth.		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE December 1978
		13. NUMBER OF PAGES 102
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

JOSEPH P. HIPPS, Major, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Error Ellipsoid   Miss Distance
Covariance Propagation
Intercept

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Errors from the intended performance of a system, composed of specialized components, could be due to any number of sub-system fluctuations. This thesis will examine the effect of these resultant errors of the mission of the system involved. In particular, errors in an interceptor's booster performance (seen as position and velocity errors at burnout) will be propagated along the interceptor's orbit to a given target point and the resultant

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73

error at the intercept point will be found. A range of intercept orbits will be examined to determine if there is a family of particular intercept orbits that give the minimum position errors at the intercept point. Conclusions will be made as to what typical trajectory should be flown against a certain type of target so as to minimize the resultant free flight error propagation to the intercept point.